



TUGAS AKHIR - TE141599

**SIMULASI KERUMUNAN MENGGUNAKAN NAVIGASI
MULTIAGEN BERBASIS RECIPROCAL VELOCITY
OBSTACLES DENGAN RELASI LEADER-FOLLOWER**

Moch Fachri
NRP 2210 100 120

Dosen Pembimbing
Mochamad Hariadi, ST., M.Sc., Ph.D.
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE141599

**CROWD SIMULATION USING MULTIAGEN NAVIGATION
BASED ON RECIPROCAL VELOCITY OBSTACLES WITH
LEADER-FOLLOWER RELATION**

Moch Fachri
NRP 2210 100 120

Advisor
Mochamad Hariadi, ST., M.Sc., Ph.D.
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

Departement of Electrical Engineering
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

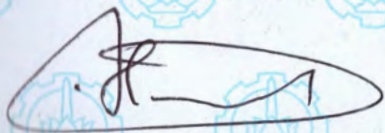
**SIMULASI KERUMUNAN MENGGUNAKAN NAVIGASI
MULTIAGEN BERBASIS *RECIPROCAL VELOCITY*
OBSTACLES DENGAN RELASI *LEADER-FOLLOWER***

TUGAS AKHIR

**Diajukan guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Komputer dan Telematika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I



Mochamad Hariadi, ST., M.Sc., Ph.D.
NIP: 196912091997031002

Dosen Pembimbing II



Dr. Supeno Mardi S N, ST., MT
NIP: 197003131995121001



SURABAYA

25 Januari, 2016

ABSTRAK

Nama Mahasiswa : Moch Fachri
Judul Tugas Akhir : Simulasi Kerumunan Menggunakan Navigasi Multiagen Berbasis *Reciprocal Velocity Obstacles* dengan Relasi *Leader-Follower*
Pembimbing : 1. Mochamad Hariadi, ST., M.Sc., Ph.D.
2. Dr. Supeno Mardi Susiki Nugroho, ST., MT.

Pada simulasi kerumunan, kepadatan agen dalam kerumunan menimbulkan masalah dalam pergerakan agen. Hal ini mengharuskan adanya metode navigasi khusus yang dapat memastikan manuver penghindaran yang tidak mengakibatkan kekacauan massal tanpa adanya komunikasi eksplisit antar agen. Untuk simulasi kerumunan dengan relasi *leader-follower*, agen *follower* diharuskan mampu mengikuti *leader* kelompoknya. *Reciprocal Velocity Obstacle* (RVO) sebagai salah satu metode navigasi multiagen tidak memerlukan komunikasi eksplisit dengan agen lain dalam penghindaran, dan bebas dari gerak osilasi yang terjadi pada *Velocity Obstacle* biasa. Diharapkan dengan memakai RVO sebagai navigasi multiagen didapatkan simulasi kerumunan dengan relasi *leader-follower* yang bebas dari tabrakan, osilasi, dan kehilangan arah.

Kata Kunci : RVO (*Reciprocal Velocity Obstacle*), Simulasi Kerumunan, Navigasi Multiagen

Halaman ini sengaja dikosongkan

ABSTRACT

Name : Moch Fachri
Title : *Crowd Simulation Using Multiagent Navigation Based on Reciprocal Velocity Obstacles with Leader-Follower Relationship*
Advisors : 1. Mochamad Hariadi, ST., M.Sc., Ph.D.
2. Dr. Supeno Mardi Susiki Nugroho, ST., MT.

In crowd simulation, agents density on crowds cause problem in agents movement. This obliges a particular navigation method that can ensure the avoidance maneuver that does not result in mass chaos without any explicit communication between agents. To simulate the crowd with leader-follower relationships, agents follower is expected to follow the group leader. Reciprocal Velocity Obstacle(RVO) as one method of multiagent navigation does not require explicit communication with other agents in avoidance, and free from oscillation movement that occur on regular Velocity Obstacle. RVO usage as multiagent navigation is expected to bring out crowd simulation with leader-follower relationship that is free of collisions, oscillation, and loss direction

Keywords : RVO (Reciprocal Velocity Obstacle), Crowd Simulation, Multiagent Navigation

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kehadiran Allah subhanahu wa ta'ala atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian dengan judul **Simulasi Kerumunan Menggunakan Navigasi Multiagen Berbasis *Reciprocal Velocity Obstacles* dengan Relasi *Leader-Follower***. Tanpa kehendak-Nya tidak mungkin penulis mampu mewujudkan penelitian ini.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Jurusan Teknik Elektro ITS, Bidang Studi Teknik Komputer dan Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Selain itu penelitian ini juga ditujukan untuk memajukan peradaban manusia di bidang teknologi komputer terutama bidang teknologi permainan. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. Ardyono Priyadi, ST., M.Eng. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.
3. Bapak Dr. Tri Arief Sardjono, ST., MT. selaku mantan Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.
4. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Mochamad Hariadi, ST., M.Sc., Ph.D. dan Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. atas bimbingan selama mengerjakan penelitian hingga memungkinkan terwujudnya hasil penelitian ini.
5. Bapak-ibu dosen pengajar Bidang Studi Teknik Komputer dan Telematika beserta dosen bidang studi lainnya, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
6. Seluruh teman-teman *B201-crew* Laboratorium Bidang Studi Teknik Komputer dan Telematika atas berbagai macam saran

DAFTAR ISI

Abstrak	i
Abstract	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xiii
1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan masalah	2
1.5 Sistematika Penulisan	3
1.6 Relevansi	4
2 TINJAUAN PUSTAKA	5
2.1 <i>Reciprocal Velocity Obstacles</i>	5
2.2 Simulasi Kerumunan	9
2.3 <i>Leader Following Behaviors</i>	10
3 DESAIN DAN IMPLEMENTASI SISTEM	13
3.1 Desain Simulasi	14
3.1.1 Skenario Pengujian	15
3.1.2 Pengambilan data	16
3.2 Desain Agen dan Navigasi	18
3.2.1 <i>Seeking</i>	18
3.2.2 <i>Steering</i>	21
3.2.3 <i>Avoiding Obstacle</i>	22
3.2.4 <i>Detecting collision</i>	24
3.3 Alur Implementasi Sistem	25
3.3.1 Pembuatan Lingkungan Simulasi	26

3.3.2	Pembuatan Agen Simulasi	33
4	PENGUJIAN DAN ANALISA	37
4.1	Pengujian Sistem Simulasi	37
4.1.1	Pengujian <i>Seeking</i>	37
4.1.2	Pengujian <i>Steering</i>	38
4.1.3	Pengujian <i>Collision Avoidance</i>	39
4.1.4	Pengujian <i>Collision Detection</i>	40
4.1.5	Pengujian ekstraksi data	42
4.2	Hasil Simulasi	42
4.2.1	<i>CrossRoad Scenario</i>	43
4.2.2	<i>Narrow Passage Scenario</i>	44
4.2.3	<i>Tour Guide Scenario</i>	49
5	PENUTUP	57
5.1	Kesimpulan	57
5.2	Saran	58
	DAFTAR PUSTAKA	59
	LAMPIRAN	61
	Biografi Penulis	81

dan masukannya beserta berbagai motivasi yang membangun dan menghancurkan.

7. Segenap civitas akademik Teknik Elektro ITS beserta Teknik Multimedia & Jaringan untuk berbagai dukungannya secara langsung maupun tidak langsung.
8. Para anggota forum.unity3d.com yang tidak bisa saya sebut satu-persatu. Terima kasih atas bantuannya dalam penelitian ini.

Kesempurnaan hanya milik Allah subhanahu wa ta'ala , untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Januari 2016

Penulis

DAFTAR GAMBAR

2.1	Konsep penghindaran tabrakan dengan <i>Velocity Obstacle</i>	5
2.2	Osilasi yang terjadi pada penghindaran dengan <i>Velocity Obstacle</i>	7
2.3	Konsep penghindaran tabrakan dengan <i>Reciprocal Velocity Obstacle</i>	8
2.4	Gerak penghindaran tabrakan dengan RVO	9
2.5	<i>Navigasi multiagen dengan RVO</i>	9
2.6	<i>Leader Following Behavior</i>	11
3.1	Contoh skenario simulasi kerumunan	16
3.2	<i>User Interface</i> informasi data simulasi	18
3.3	<i>Flowchart</i> alur pergerakan agen setiap <i>cycle</i>	19
3.4	<i>Flowchart seeking</i>	20
3.5	<i>Flowchart steering</i>	23
3.6	<i>Flowchart collision avoidance</i>	24
3.7	Desain RVO untuk simulasi	25
3.8	<i>Flowchart collision detection</i>	26
3.9	Desain lingkungan simulasi <i>crossroad scenario</i>	27
3.10	Area <i>walkable</i> (biru) untuk <i>crossroad scenario</i>	28
3.11	Desain lingkungan simulasi <i>narrow-passage scenario</i>	29
3.12	Area <i>walkable</i> untuk <i>narrow-passage scenario</i>	30
3.13	Denah pameran	31
3.14	<i>Layout</i> skenario <i>tour guide</i>	31
3.15	area <i>walkable</i> pada <i>tourguide scenario</i>	32
3.16	Agen simulasi kerumunan	34
3.17	Lebar agen dan separasi antar agen	35
3.18	<i>Collider</i> yang ditanam pada tubuh agen	36
4.1	Hasil <i>seeking</i> agen <i>leader & follower</i>	38
4.2	Pengejaran oleh agen <i>follower</i> yang tertinggal	39
4.3	<i>Steering</i> agen <i>follower</i> yang berada di depan <i>leader</i> -nya	40
4.4	Hasil penghindaran tabrakan dengan menggunakan RVO	41
4.5	Hasil pengujian deteksi tabrakan	41

4.6	Grafik jumlah tabrakan yang terjadi pada simulasi <i>crossroad scenario</i>	44
4.7	Grafik jumlah frame pada simulasi <i>crossroad scenario</i>	45
4.8	Grafik <i>collision per frame</i> untuk simulasi <i>crossroad scenario</i>	45
4.9	Grafik waktu simulasi <i>crossroad scenario</i>	46
4.10	Grafik rata-rata <i>frame rate</i> untuk simulasi <i>crossroad scenario</i>	46
4.11	Grafik peningkatan tabrakan agen untuk simulasi <i>crossroad scenario</i>	47
4.12	Grafik penurunan waktu simulasi <i>crossroad scenario</i>	47
4.13	Grafik jumlah tabrakan yang terjadi pada simulasi <i>narrow passage scenario</i>	48
4.14	Grafik jumlah frame pada simulasi <i>narrow passage scenario</i>	49
4.15	Grafik <i>collision per frame</i> untuk simulasi <i>narrow passage scenario</i>	49
4.16	Grafik waktu simulasi <i>narrow passage scenario</i> . . .	50
4.17	Grafik rata-rata <i>frame rate</i> untuk simulasi <i>narrow passage scenario</i>	50
4.18	Grafik peningkatan tabrakan agen untuk simulasi <i>narrow passage scenario</i>	51
4.19	Grafik penurunan waktu simulasi <i>narrow passage scenario</i>	51
4.20	Grafik jumlah tabrakan yang terjadi pada simulasi <i>tour guide scenario</i>	53
4.21	Grafik jumlah frame pada simulasi <i>tour guide scenario</i>	53
4.22	Grafik <i>collision per frame</i> untuk simulasi <i>tour guide scenario</i>	54
4.23	Grafik waktu simulasi <i>tour guide scenario</i>	54
4.24	Grafik rata-rata <i>frame rate</i> untuk simulasi <i>tour guide scenario</i>	55
4.25	Grafik peningkatan tabrakan agen untuk simulasi <i>tour guide scenario</i>	55
4.26	Grafik peningkatan waktu simulasi <i>tour guide scenario</i>	56
E1	<i>Heatmap</i> skenario <i>crossroad</i> 120 agen dengan relasi .	68
E2	<i>Heatmap</i> skenario <i>crossroad</i> 130 agen dengan relasi .	69

E3	<i>Heatmap</i> skenario <i>crossroad</i> 140 agen dengan relasi .	69
E4	<i>Heatmap</i> skenario <i>crossroad</i> 150 agen dengan relasi .	69
E5	<i>Heatmap</i> skenario <i>crossroad</i> 160 agen dengan relasi .	69
E6	<i>Heatmap</i> skenario <i>crossroad</i> 170 agen dengan relasi .	70
E7	<i>Heatmap</i> skenario <i>crossroad</i> 180 agen dengan relasi .	70
E8	<i>Heatmap</i> skenario <i>crossroad</i> 190 agen dengan relasi .	70
E9	<i>Heatmap</i> skenario <i>crossroad</i> 200 agen dengan relasi .	70
E10	<i>Heatmap</i> skenario <i>crossroad</i> 140 agen tanpa relasi . .	70
E11	<i>Heatmap</i> skenario <i>crossroad</i> 150 agen tanpa relasi . .	71
E12	<i>Heatmap</i> skenario <i>crossroad</i> 160 agen tanpa relasi . .	71
E13	<i>Heatmap</i> skenario <i>crossroad</i> 170 agen tanpa relasi . .	71
E14	<i>Heatmap</i> skenario <i>crossroad</i> 180 agen tanpa relasi . .	71
E15	<i>Heatmap</i> skenario <i>crossroad</i> 190 agen tanpa relasi . .	72
E16	<i>Heatmap</i> skenario <i>crossroad</i> 200 agen tanpa relasi . .	72
E17	<i>Heatmap</i> skenario <i>narrow passage</i> 128 agen dengan relasi	72
E18	<i>Heatmap</i> skenario <i>narrow passage</i> 136 agen dengan relasi	73
E19	<i>Heatmap</i> skenario <i>narrow passage</i> 144 agen dengan relasi	73
E20	<i>Heatmap</i> skenario <i>narrow passage</i> 152 agen dengan relasi	73
E21	<i>Heatmap</i> skenario <i>narrow passage</i> 160 agen dengan relasi	74
E22	<i>Heatmap</i> skenario <i>narrow passage</i> 168 agen dengan relasi	74
E23	<i>Heatmap</i> skenario <i>narrow passage</i> 176 agen dengan relasi	74
E24	<i>Heatmap</i> skenario <i>narrow passage</i> 184 agen dengan relasi	75
E25	<i>Heatmap</i> skenario <i>narrow passage</i> 192 agen dengan relasi	75
E26	<i>Heatmap</i> skenario <i>narrow passage</i> 200 agen dengan relasi	75
E27	<i>Heatmap</i> skenario <i>narrow passage</i> 136 agen tanpa relasi	76
E28	<i>Heatmap</i> skenario <i>narrow passage</i> 144 agen tanpa relasi	76
E29	<i>Heatmap</i> skenario <i>narrow passage</i> 152 agen tanpa relasi	76
E30	<i>Heatmap</i> skenario <i>narrow passage</i> 160 agen tanpa relasi	77

E31	<i>Heatmap</i> skenario <i>narrow passage</i> 168 agen tanpa relasi	77
E32	<i>Heatmap</i> skenario <i>narrow passage</i> 176 agen tanpa relasi	77
E33	<i>Heatmap</i> skenario <i>narrow passage</i> 184 agen tanpa relasi	78
E34	<i>Heatmap</i> skenario <i>narrow passage</i> 192 agen tanpa relasi	78
E35	<i>Heatmap</i> skenario <i>narrow passage</i> 200 agen tanpa relasi	78
E36	<i>Collision heatmap legend</i>	79

DAFTAR TABEL

3.1	Titik awal dan akhir agen <i>leader</i> dalam <i>crossroad scenario</i>	28
3.2	Koordinat titik yang ditunjukkan gambar 3.11	28
3.3	Titik mulai dan <i>waypoint</i> agen <i>leader tour guide scenario</i>	33
4.1	Hasil pengujian fungsi pengambilan data	42
A1	Data simulasi <i>crossroad scenario leader-follower</i> . . .	61
A2	Data simulasi <i>crossroad scenario</i> tanpa relasi	61
A3	Data perbandingan RVO dengan relasi <i>leader-follower</i> dan tanpa relasi pada <i>crossroad scenario</i>	62
B1	Data simulasi <i>narrow passage leader-follower</i>	63
B2	Data simulasi <i>narrow passage</i> tanpa relasi	63
B3	Data perbandingan RVO dengan relasi <i>leader-follower</i> dan tanpa relasi pada <i>narrow passage scenario</i>	64
C1	Data simulasi <i>tour guide leader-follower</i>	65
C2	Data simulasi <i>tour guide</i> tanpa relasi	66
C3	Data perbandingan RVO dengan relasi <i>leader-follower</i> dan tanpa relasi pada <i>crossroad scenario</i>	67
D	Data jumlah agen yang tersesat pada penggunaan relasi <i>leader - follower</i>	67

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Simulasi kerumunan merupakan proses permodelan gerakan dari sekumpulan entitas bergerak yang disebut agen dalam jumlah besar dengan kepadatan tinggi. Dengan simulasi kerumunan, situasi tempat ramai di dunia seperti bandar udara, stadion, stasiun, dan semisalnya bisa dimodelkan secara virtual. Secara umum, simulasi kerumunan memiliki kerumunan agen dengan masing-masing agen pergi ke titik tujuan tertentu pada dunia simulasi.

Masalah yang muncul dalam simulasi kerumunan akibat kepadatan agen dalam kerumunan sulit dihindari, sehingga diperlukan metode navigasi khusus yang dapat memastikan manuver penghindaran agen kerumunan tidak mengakibatkan kekacauan massal. Kemampuan navigasi masing-masing agen berperan penting, karena navigasi agen menentukan gerakan seluruh kelompok menuju titik tujuannya dengan jarak optimal secara aman, dalam artian tidak terjadi kemacetan dan tabrakan baik antar agen maupun rintangan yang menghadang. Kesalahan navigasi satu agen saja akan mengakibatkan seluruh kerumunan tidak dapat bergerak. Karena itu dikembangkan berbagai macam metode navigasi multiagen agar agen-agen dapat bergerak dalam kerumunan berkepadatan tinggi dengan aman.

Reciprocal Velocity Obstacle (RVO) sebagai salah satu metode navigasi multiagen memiliki banyak fitur yang masih dapat dikembangkan. Salah satunya adalah penggunaan RVO untuk navigasi agen-agen yang memiliki relasi pemimpin dan pengikut (*leader & follower*). Penelitian mengenai navigasi multiagen dengan RVO masih terfokus dalam skenario tiap agen memiliki titik tujuannya masing-masing tanpa ada relasi selain relasi kelompok[1][2][3][4][5][6]. Oleh karena itu diperlukan penelitian keandalan metode RVO pada na-

vigasi multiagen dengan relasi *leader & follower*. Hasil penelitian ini bisa diterapkan pada simulasi kerumunan yang membutuhkan skenario kerumunan agen *follower* mengikuti seorang agen *leader*.

1.2 Permasalahan

Metode navigasi multiagen pada simulasi kerumunan dengan relasi pemimpin dan pengikut (*leader & follower*) perlu memiliki manuver penghindaran terhadap agen lain, dan rintangan yang menghadang. Simulasi kerumunan juga mensyaratkan tidak adanya komunikasi eksplisit antar agen sehingga tidak ada koordinasi terpusat dalam menentukan gerak navigasi agen. Disaat bersamaan agen harus mempertahankan kecepatannya terhadap agen pemimpin yang diikuti. Seringkali kesalahan navigasi agen menyebabkan agen dalam simulasi terjebak dalam kemacetan bahkan tabrakan, tidak mampu menentukan arah tujuan, serta melakukan gerak penghindaran dengan osilasi. Semua itu mengakibatkan agen tidak mampu mengikuti pemimpin dan secara global akan mengganggu navigasi agen lain.

1.3 Tujuan

Navigasi multiagen menggunakan RVO tidak memerlukan komunikasi eksplisit dengan agen lain dalam penghindaran, dan bebas dari gerak osilasi seperti yang terjadi pada *Velocity Obstacle* biasa[1]. Karena itu RVO dapat mengatasi masalah penghindaran rintangan tanpa koordinasi terpusat di simulasi kerumunan. Penelitian ini bertujuan untuk mendapatkan data keefektifan navigasi multiagen menggunakan RVO pada simulasi kerumunan yang memiliki relasi pengikut dan pemimpin bila dibandingkan dengan tanpa relasi. Dari hasil simulasi yang didapat, diharapkan seluruh agen dengan relasi pada simulasi bisa mencapai tujuan akhir dengan tingkat tabrakan, dan waktu yang lebih baik sehingga dapat dimanfaatkan pada simulasi kerumunan dengan berbagai skenario yang berbeda.

1.4 Batasan masalah

Batasan masalah pada penelitian ini adalah:

1. Pergerakan agen hanya pada bidang dimensi dua di satu lantai.
2. Jumlah agen dibatasi hingga 200 agen.

3. Jumlah kelompok kerumunan dibatasi sebanyak empat kelompok dengan masing masing kelompok memiliki satu pemimpin utama.
4. Ukuran dan tipe agen tidak variasi.

1.5 Sistematika Penulisan

Laporan penelitian tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan
Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. BAB II Dasar Teori
Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori-teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait *reciprocal velocity obstacle*, navigasi multiagen ,simulasi kerumunan, dan teori-teori penunjang lainnya.
3. BAB III Perancangan Sistem dan Implementasi
Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Guna mendukung itu digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan penelitian tugas akhir.
4. BAB IV Pengujian dan Analisa
Bab ini menjelaskan tentang pengujian yang dilakukan terhadap sistem dalam penelitian ini dan menganalisa sistem. Spesifikasi perangkat keras dan perangkat lunak yang diuji juga disebutkan dalam bab ini. Sehingga ketika akan dikembangkan lebih jauh, spesifikasi perlengkapannya bisa dipenuhi dengan mudah tanpa harus melakukan ujicoba perangkat lunak maupun perangkat keras lagi.
5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk mengembangkan lebih lanjut juga dituliskan pada bab ini.

1.6 Relevansi

Penelitian mengenai simulasi kerumunan merupakan salah satu bidang penelitian dalam teknologi permainan (game technology) untuk navigasi virtual agent dalam permainan dengan jumlah agen besar di ruang terbatas. Selain dalam bidang teknologi permainan, simulasi ini dapat diimplementasikan dalam bidang robotika untuk navigasi robot. Perkembangan penelitian ini dapat digunakan untuk riset dan penelitian mengenai alternatif lain dalam navigasi multiagen terutama untuk simulasi kerumunan.

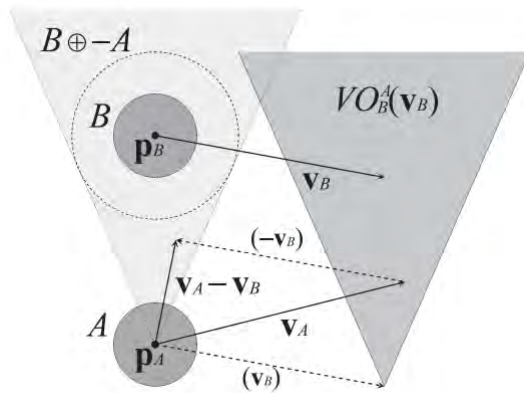
BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 *Reciprocal Velocity Obstacles*

Reciprocal Velocity Obstacles (RVO) adalah konsep untuk *local reactive collision avoidance* yang digunakan dalam navigasi multi-agen yang tiap agen melakukan navigasi secara independent tanpa ada komunikasi secara eksplisit[1]. RVO memperbaiki konsep Velocity Obstacle[7] yang sebelumnya dengan tujuan menghasilkan gerakan agen yang aman dari tabrakan dengan agen lain dan rintangan, serta bebas dari osilasi yang terjadi pada *Velocity Obstacle*.



Gambar 2.1: Konsep penghindaran tabrakan dengan *Velocity Obstacle*.
Sumber gambar:[1]

Gambar 2.1 menunjukkan konsep penghindaran tabrakan dengan *Velocity Obstacle* yang dijelaskan oleh Jur van der Berg, dkk[1] sebagai berikut. Misalkan agen A adalah agen sedang bertranslasi

di bidang dengan titik referensinya pada \mathbf{P}_A , dan B adalah rintangan planar(bergerak) dengan titik referensi pada \mathbf{P}_B . *Velocity Obstacle* $VO_B^A(\mathbf{v}_B)$ dari rintangan B ke agent A adalah himpunan yang terdiri dari semua kecepatan \mathbf{v}_A untuk A yang akan menghasilkan tabrakan pada suatu momen waktu bersamaan dengan rintangan B bergerak dengan kecepatan \mathbf{v}_B .

Velocity Obstacle dari agen A terhadap rintangan B tersebut dapat didefinisikan secara geometri sebagai:

$$A \oplus B = \{\mathbf{a} + \mathbf{b} | \mathbf{a} \in A, \mathbf{b} \in B\}, -A = \{-\mathbf{a} | \mathbf{a} \in A\} \quad (2.1)$$

$A \oplus B$ adalah penjumlahan Minkowski dari dua objek A dan B dimana $-A$ adalah refleksi objek A terhadap titik referensinya. Misal $\lambda(\mathbf{p}, \mathbf{v})$ menyatakan *ray* yang bermula di \mathbf{p} dan bergerak ke arah \mathbf{v} :

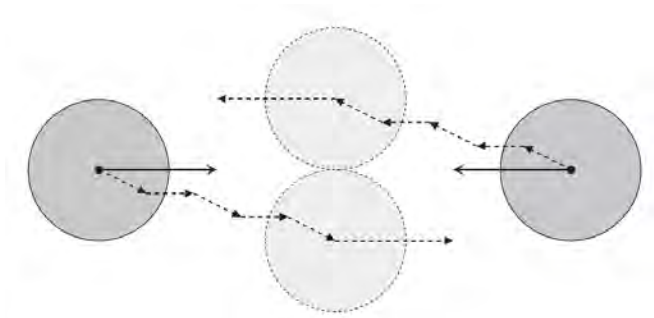
$$\lambda(\mathbf{p}, \mathbf{v}) = \{\mathbf{p} + t\mathbf{v} | t \geq 0\} \quad (2.2)$$

Jika *ray* yang bermula pada \mathbf{p}_A dan mengarah ke kecepatan relatif dari A dan B ($\mathbf{v}_A - \mathbf{v}_B$) memotong jumlah Minkowski dari B dan A yang berpusat di \mathbf{p}_B , Velocity Obstacle \mathbf{v}_A berada didalam *Velocity Obstacle* dari B. Sehingga Velocity Obstacle B ke A didefinisikan sebagai:

$$VO_B^A(\mathbf{v}_B) = \{\mathbf{v}_A | \lambda(\mathbf{p}_A, \mathbf{v}_A - \mathbf{v}_B) \cap B \oplus -A \neq \emptyset\} \quad (2.3)$$

Ini berarti jika $\mathbf{v}_A \in VO_B^A(\mathbf{v}_B)$, A dan B akan bertabrakan pada suatu titik waktu. Jika \mathbf{v}_A diluar *Velocity Obstacle* dari B, kedua objek tiada akan pernah bertabrakan. Jika \mathbf{v}_A ada pada perbatasan *Velocity Obstacle*, maka A akan bersentuhan dengan B pada suatu momen waktu.

Konsep *Velocity Obstacle* dapat digunakan untuk navigasi multiagen dengan tiap agen menganggap agen lain sebagai rintangan bergerak dan memilih kecepatan untuk dirinya sendiri yang berada diluar *Velocity Obstacle* yang diinduksikan dari agen lain. Namun, pendekatan ini menghasilkan pergerakan osilasi yang tidak diinginkan.



Gambar 2.2: Osilasi yang terjadi pada penghindaran dengan *Velocity Obstacle*. Sumber gambar:[1]

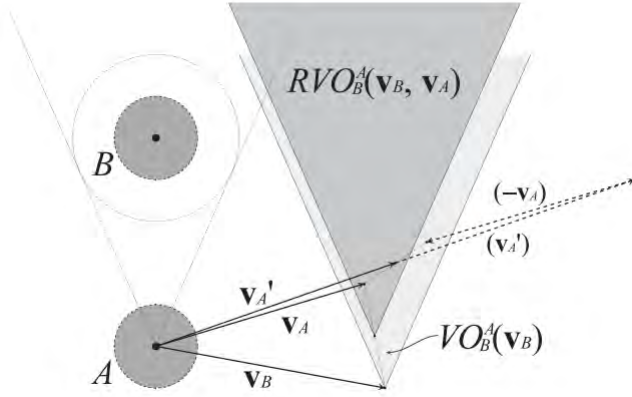
Misalkan pada situasi yang ditunjukkan gambar 2.2, dua agen A dan B bergerak dengan kecepatan \mathbf{v}_A dan \mathbf{v}_B bersamaan sehingga $\mathbf{v}_A \in VO_B^A(\mathbf{v}_B)$ dan $\mathbf{v}_B \in VO_A^B(\mathbf{v}_A)$. Karenanya, jika kedua agen tetap melanjutkan kecepatan yang telah ada akan berakibat pada tabrakan antara keduanya. Untuk menghindarinya, agen A akan mengubah kecepatannya menjadi \mathbf{v}_A yang berada diluar *Velocity Obstacle* B ($\mathbf{v}_A \notin VO_A^B(\mathbf{v}_B)$). Disaat bersamaan agen B juga mengubah kecepatannya menjadi \mathbf{v}_B yang berada diluar *Velocity Obstacle* A ($\mathbf{v}_B \notin VO_B^A(\mathbf{v}_A)$).

Masalah muncul ketika 2 agen berada pada kecepatan \mathbf{v}_A dan \mathbf{v}_B , kecepatan lama (\mathbf{v}_A dan \mathbf{v}_B) sudah berada diluar *Velocity Obstacle* A dan B. Kedua agen akan kembali pada kecepatan yang lama karena dengan kecepatan yang lama, jarak menuju titik tujuan masing-masing agen adalah yang terpendek. Pada siklus selanjutnya kedua agen dengan kecepatan lama akan bertabrakan sehingga kecepatan mereka berdua akan berubah lagi menjadi \mathbf{v}_A dan \mathbf{v}_B . Hal ini berulang hingga kedua agen berpapasan. Perubahan kecepatan terus menerus dari \mathbf{v}_A dan \mathbf{v}_B ke \mathbf{v}_A dan \mathbf{v}_B kemudian sebaliknya inilah yang menimbulkan gerak osilasi yang ditunjukkan di gambar 2.2.

Untuk menghindari gerak osilasi inilah, Jur van der Berg, dkk[1] merumuskan metode baru yang dikembangkan dari *Velocity Obstacle* yaitu *Reciprocal Velocity Obstacle* (RVO). Ide dasarnya sederhana

na, Kecepatan(*velocity*) baru yang dipilih untuk menghindari tabrakan merupakan rerata dari kecepatan agen sekarang dan kecepatan yang berada diluar *Velocity Obstacle* agen lain. RVO didefinisikan sebagai:

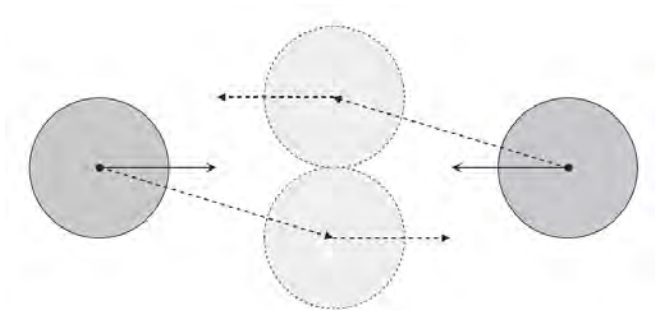
$$RVO_B^A(\mathbf{v}_B, \mathbf{v}_A) = \{\mathbf{v}'_A | \mathbf{v}'_A - \mathbf{v}_A \in VO_B^A(\mathbf{v}_B)\} \quad (2.4)$$



Gambar 2.3: Konsep penghindaran tabrakan dengan *Reciprocal Velocity Obstacle*. Sumber gambar:[1]

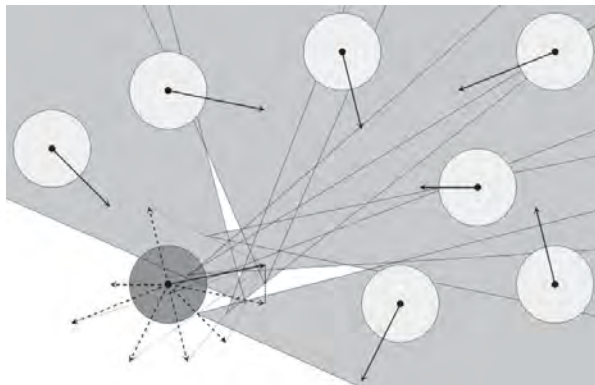
Gambar 2.3 menunjukkan *Reciprocal Velocity Obstacle* $RVO_B^A(\mathbf{v}_B, \mathbf{v}_A)$ dari agen B ke agen A berisi semua kecepatan agen A yang merupakan rerata kecepatan \mathbf{v}_A dan kecepatan yang berada didalam *Velocity Obstacle* $VO_B^A(\mathbf{v}_B)$ agen B. Kemudian RVO_B^A secara geometris dapat diinterpretasikan sebagai *Velocity Obstacle* VO_B^A yang ditranslasikan supaya nilai puncaknya berada di $\frac{\mathbf{v}_A + \mathbf{v}_B}{2}$. Gerakan penghindaran yang dihasilkan dapat dilihat pada gambar 2.4.

Untuk navigasi multiagen, agen akan memilih *velocity* berdasarkan *valid velocity* yang dihasilkan dari RVO yaitu *velocity* yang berada diluar gabungan seluruh area *reciprocal velocity obstacles* dan yang paling mendekati *velocity* yang diinginkan. Dapat dilihat



Gambar 2.4: Gerak penghindaran tabrakan dengan RVO. Sumber gambar:[1]

pada gambar 2.5, area gerak yang diarsir gelap menunjukkan *invalid velocity* untuk agen bergerak berwarna gelap yang merupakan *union* dari RVO yang dihasilkan agen berwarna cerah.



Gambar 2.5: Navigasi multiagen dengan RVO. Sumber gambar:[1]

2.2 Simulasi Kerumunan

Kerumunan adalah sekumpulan individu di dalam lingkungan fisik yang sama, berbagi tujuan yang sama[8]. Simulasi kerumunan dipakai untuk memproduksi gerakan dan menampilkan perilaku

sekumpulan kerumunan yang alami[9]. Ada tiga pendekatan yang luas untuk simulasi kerumunan[8] yaitu:

1. Pendekatan fluida. Dapat diobservasi bahwa gerakan kerumunan pada level makroskopik mirip dengan aliran fluida. Telah ada percobaan yang berhasil memodelkan agen menggunakan hukum fisika seputar dinamika fluida.
2. Pendekatan otomata seluler yaitu suatu pendekatan sistem diskret dan dinamik yang memodelkan suatu struktur sel yang mendasarkan status suatu sel pada status sel yang berhubungan langsung dengannya.
3. Pendekatan Partikel, kemungkinan pendekatan ini yang paling sering digunakan pada mayoritas percobaan yang dijelaskan di berbagai literatur. Sering disebut sebagai pendekatan partikulat atau atomik. Dalam skema ini setiap agen dianggap sebagai satu entitas dan interaksi antar agen dimodelkan secara individual berdasarkan hukum fisik atau sosial.

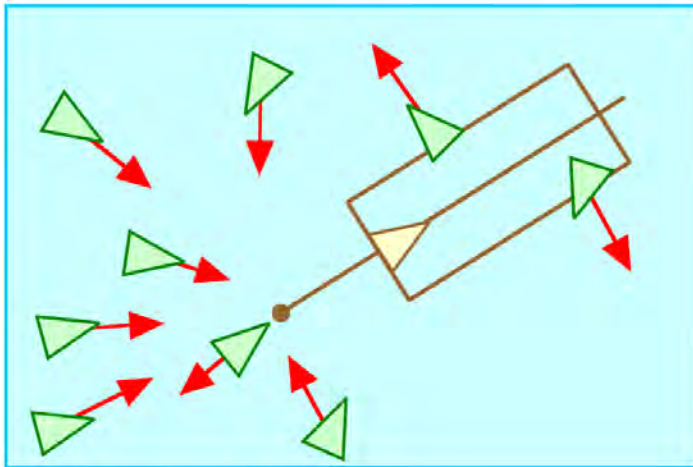
Simulasi kerumunan ini dapat didekati dalam level makroskopik dan mikroskopik. Pada level makroskopik, fokus utama berada di kerumunan massa orang-orang bergerak dengan gerakan yang realistis, tanpa perlu terfokus oleh gerakan individual. Namun pada level mikroskopik, pada tiap-tiap individu agen bergerak dalam pergerakan yang meyakinkan perlu difokuskan. Pendekatan pada level makroskopik maupun mikroskopik tergantung pada aplikasi dari simulasi yang akan dibuat. Secara umum pada level makroskopik metode pendekatan berbasis fluida adalah yang paling baik, sedangkan pada level mikroskopis metode yang sering dipakai adalah metode pendekatan berbasis partikel.

2.3 *Leader Following Behaviors*

Leader following behavior adalah *steering behavior* untuk gerak *boids* dimana *boids* diharuskan bergerak mengikuti *leader* yang telah ditentukan. *Boids* ini disebut sebagai *follower*. Berdasarkan *steering behavior* untuk *boids* yang didesain oleh Craig W. Reynolds[10], *Leader Following Behaviors* dibentuk dari dua macam *behavior* yaitu *arrival* dan *separation*. Kedua *behavior* tersebut terbentuk karena secara umum, *follower* ingin tetap bergerak dekat dengan *leader*

tanpa menghalangi pergerakannya disertai dengan penghindaran tabrakan dengan *follower* lain seperti pada gambar 2.6.

Arrival behavior mengakibatkan *follower* berusaha bergerak ke titik tujuannya yaitu posisi *leader* dan berusaha melambat ketika jarak dengan leader semakin mendekat. Sebaliknya, semakin tertinggal agen tersebut dengan *leader* maka kelajuan agen tersebut akan meningkat untuk mengejar ketertinggalannya. Jika *follower* berada di area rectangular di depan *leader*, maka follower akan melakukan *steering* menghindari jalur *leader* sebelum melanjutkan *arrival behavior*. Adapun *separation* diimplementasikan untuk mencegah para *follower* bergerombol satu sama lain. Hal ini sebagai tindakan pencegahan terjadinya *collision* antar *follower*



Gambar 2.6: *Leader Following Behavior*. Sumber gambar:[10]

Halaman ini sengaja dikosongkan

BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Tujuan yang ingin dicapai dari penelitian ini adalah mendapatkan data keefektifan navigasi multiagen menggunakan RVO pada simulasi kerumunan yang memiliki relasi pengikut dan pemimpin. Sebelum melakukan pengukuran performa, infrastruktur dan sistem dari simulasi kerumunan harus dibangun terlebih dahulu.

Infrastruktur yang pertama dibuat adalah desain simulasi. Desain simulasi dibuat berdasarkan skenario yang telah banyak dipakai sebagai pengujian performansi navigasi agen kerumunan. Skenario simulasi yang dipakai untuk pengujian memakai skenario *crossroad* dan *narrow passage* yang pernah dipakai oleh Jur van den Berg pada penelitiannya mengenai RVO[1][3] serta skenario *tour guide*. Pada *crossroad* dan *narrow passage scenario* yang sudah diujikan untuk RVO, masih belum diimplementasikan adanya relasi *leader & follower*[1][3]. Tiap agen masih bergerak menuju titik tujuannya masing-masing tanpa adanya relasi antar agen. Pada penelitian ini akan ditambahkan relasi *leader & follower* sehingga agen *follower* perlu mengikuti masing-masing *leader*-nya, dan *leader* tersebut yang akan bergerak ke titik tujuan kelompok kerumunan.

Untuk simulasi *tour guide scenario*, desain simulasi ini akan mengambil permodelan dari dunia nyata yaitu tempat tur. Nantinya pada skenario ini terdapat agen *follower* yang menjadi pengunjung tempat tur dan pemandu sebagai *leader* yang akan diikuti para pengunjung. Pada skenario ini, *leader* memiliki beberapa titik tujuan tidak seperti skenario sebelumnya yang hanya memiliki satu titik tujuan. Dari seluruh skenario yang disimulasikan, akan didapatkan data-data yang menunjukkan performansi RVO pada simulasi kerumunan.

Setelah desain dunia simulasi dibuat, yang selanjutnya adalah mendesain agen yang akan ditempatkan di skenario simulasi beserta

navigasinya. Agen pada simulasi dibentuk dari objek berbentuk kapsul seperti pada gambar 3.16. Dalam navigasinya, agen harus menemukan kemana destinasinya beserta cara agen tersebut menuju destinasinya. Proses kalkulasi gerak agen dilakukan setiap siklus hingga simulasi selesai.

Proses awal pada tiap siklus adalah pencarian destinasi yang disebut proses *seeking* dimana agen *follower* mencari *leader* kelompoknya sebagai titik tujuan, dan *leader* mencari titik rute perjalanan yang telah ditentukan. Setelah ditemukan tujuan seluruh agen, proses berikutnya adalah pengarahan agen menuju titik tujuannya atau disebut *steering*, keluaran dari *steering* menghasilkan *velocity* referensi yang akan diuji validitasnya pada proses *collision avoidance* dimana *velocity* agen untuk siklus tersebut harus terbebas dari tabrakan dengan menggunakan algoritma RVO. Bila *velocity* yang dihasilkan *steering* dapat menyebabkan *collision* maka dipilih *velocity* baru sesuai dengan aturan RVO, yaitu *velocity* yang tidak berada di daerah *velocity obstacle* yang terdekkan dengan *velocity* referensi. Proses terakhir adalah pendeteksian terjadinya tabrakan untuk mendapatkan data keberhasilan RVO dalam mencegah terjadinya tabrakan baik antar agen maupun dengan rintangan. keempat proses ini akan dilakukan terus menerus setiap siklusnya selama simulasi berjalan.

Lingkungan simulasi virtual serta agen simulasi dibuat menggunakan Unity *game engine*. Fungsi utamanya adalah untuk membangun lingkungan simulasi virtual serta pengimplementasian navigasi multiagen pada kerumunan agen yang akan digunakan untuk skenario pengujian. Dengan memakai *game engine* ini juga data-data dari simulasi diekstraksi.

3.1 Desain Simulasi

Untuk mendapatkan data keefektifan navigasi multiagen menggunakan RVO pada simulasi kerumunan yang memiliki relasi *leader & follower*, maka perlu didesain skenario simulasi untuk menguji algoritma navigasi agen tersebut beserta ekstraksi datanya.

3.1.1 Skenario Pengujian

Dalam penelitian ini ada tiga skenario simulasi yang akan digunakan. Dua skenario pertama adalah skenario dasar yang banyak digunakan pada penelitian seputar navigasi multiagen. Skenario ketiga adalah skenario yang mengambil model lingkungan dunia nyata.

1. *Crossroad Scenario*[3]

Skenario ini adalah skenario yang meniru situasi kerumunan penyeberangan jalan, dimana pada skenario ini terdapat dua kerumunan agen yaitu kerumunan A dan B. Kerumunan agen A akan berjalan menuju lokasi awal kerumunan NPC B, sedangkan kerumunan agen B akan berjalan menuju ke tempat awal kerumunan agen A. Dalam skenario ini kerumunan agen A dan B akan bertemu dan harus menghindar satu sama lain untuk mencegah terjadinya tabrakan.

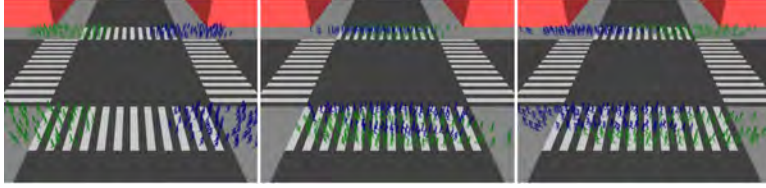
2. *Narrow Passage Scenario*[1]

Pada skenario ini terdapat empat kerumunan agen yaitu kerumunan A,B,C, dan D. Kerumunan agen A akan berjalan menuju lokasi awal kerumunan agen C, kerumunan agen B akan berjalan menuju ke tempat awal kerumunan agen D, kerumunan agen C akan berjalan menuju ke tempat awal kerumunan agen A, dan kerumunan agen D akan berjalan menuju ke tempat awal kerumunan agen A. Semua kerumunan akan melewati rintangan lingkungan berupa empat buah kotak yang akan menyebabkan empat kerumunan ini harus melewati jalur sempit yang terbentuk dari empat buah kotak ini. Semua kerumunan agen akan bertemu sehingga selain harus melewati rintangan lingkungan, semua NPC juga harus menghindari NPC lain.

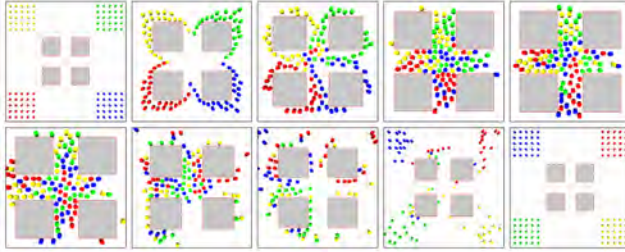
3. *Tour Guide Scenario*

Pada skenario ini terdapat empat kerumunan agen yang mengikuti pemimpinnya dalam mengelilingi tempat tur dengan spot tertentu yang tiap kerumunan memiliki jalur tur yang berbeda. Berbeda dengan skenario sebelumnya yang mana setiap *leader* hanya memiliki satu titik tujuan, pada skenario ini ada beberapa *waypoint* yang harus dilalui *leader*. Skenario

simulasi ini mengambil model dari denah pameran alutsista TNI(Tentara Nasional Indonesia) yang diselenggarakan tanggal 5 & 6 September 2015 di kampus Universitas Airlangga. Selain sebagai pengujian, skenario ini juga sebagai contoh pengimplementasian RVO dengan relasi *leader & follower*.



(a) *Crossroad scenario*



(b) *Narrow-passage scenario*

Gambar 3.1: Contoh skenario simulasi kerumunan pada penelitian [1] dan [3]

3.1.2 Pengambilan data

Dalam simulasi ini ada data yang akan diambil dari *game engine* yang digunakan. Data ini dipakai untuk menganalisa performa dari simulasi kerumunan yang memiliki relasi *leader & follower* dengan penggunaan RVO sebagai navigasi multiagennya. Data tersebut adalah:

1. *Total frame*

Merupakan data yang menunjukkan jumlah *frame* yang ada selama simulasi. Tiap *frame* menunjukkan *cycle* perhitungan pergerakan agen. Total *frame* simulasi didapat dengan menghitung pemanggilan fungsi *update per frame* yang selalu

dipanggil pada Unity setiap *frame game* berjalan.

$$\Sigma Frame = \Sigma Function_{update()} \quad (3.1)$$

2. *Simulation time*

Data ini merepresentasikan waktu yang diperlukan untuk melakukan simulasi. Simulasi berhenti ketika semua agen telah sampai pada titik tujuannya atau terjadi *deadlock* sehingga mengakibatkan agen yang belum mencapai titik tujuan tidak dapat melanjutkan perjalanan. Data *simulation time* didapatkan dengan mengambil data pewaktu internal Unity, dimana waktu simulasi disamakan dengan waktu berjalannya permainan.

$$\Sigma Simulation Time = \Sigma Time_{\Delta Time} \quad (3.2)$$

3. *Total collision*

Sebagaimana telah disebutkan pada bagian 3.2.3 , terdapat data jumlah terjadinya tabrakan agen selama simulasi. Dari data ini dapat dianalisis keefektifan RVO dalam penghindaran tabrakan pada berbagai kepadatan kerumunan. Data ini diambil dari akumulasi *collision* yang dideteksi oleh *collider* agen.

$$\Sigma Collision = \Sigma Function_{OnCollisionEnter()} \quad (3.3)$$

4. *Average frame rate*

Merupakan nilai yang menunjukkan rerata dari jumlah *frame* persatuan detik. Didapatkan dengan membagi jumlah *frame* selama simulasi berjalan dengan *simulation time* dan memiliki satuan FPS(*Frame persecond*).

$$\widetilde{FPS} = \frac{\Sigma Frame}{\Sigma Simulation Time} \quad (3.4)$$

5. *Collision perframe*

Nilai ini menunjukkan jumlah *collision* untuk tiap *frame*. Didapatkan dengan membagi jumlah *collision* dengan jumlah

frame dan memiliki satuan CPF(*Collision perframe*).

$$CPF = \frac{\Sigma Collision}{\Sigma Frame} \quad (3.5)$$

6. *Total loss agents*

Merupakan data yang menunjukkan jumlah agen *follower* yang telah kehilangan arah, yang berarti agen tersebut tidak dapat menemukan leadernya.

$$\Sigma Loss\ Agents = \Sigma Follower_{loss=true} \quad (3.6)$$

Semua data yang diambil akan ditampilkan pada *user interface* simulasi seperti yang ditunjukkan pada gambar 3.2.

Frame Rate	Total Collision	Total Frame	Collision/Frame	Time	Strayed Agents
60.4 FpS	19	1520	0.01250823	25.3202	0

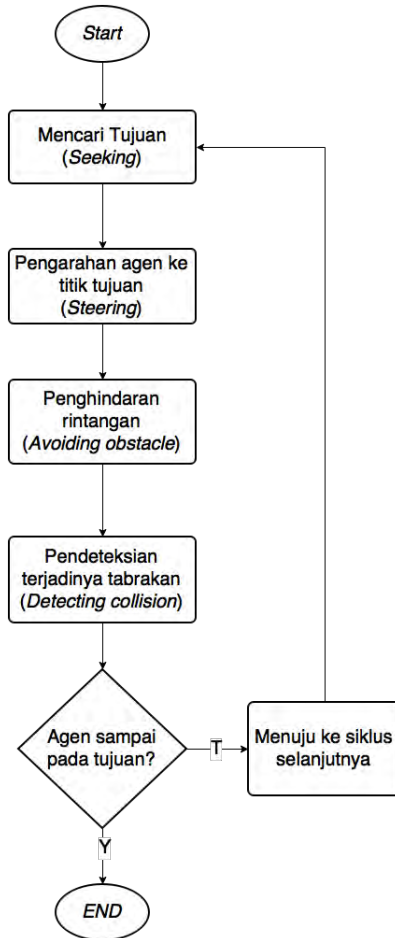
Gambar 3.2: *User Interface* informasi data simulasi. Dari kiri ke kanan: *frame rate*, jumlah tabrakan yang terjadi, Jumlah *frame*, jumlah tabrakan per siklus *frame*, waktu simulasi, jumlah agen tersesat.

3.2 Desain Agen dan Navigasi

Rancangan dasar dari pergerakan agen di simulasi ini ditunjukkan pada gambar 3.3. Kalkulasi pergerakan agen terdiri dari empat tahap untuk setiap siklus berjalannya simulasi yang pada Unity dikalkulasi setiap *frame*. Pergerakan agen dibangun menggunakan sistem navigasi yang telah disediakan Unity. Dalam pergerakan agen ini ada dua hal yang menjadi perhatian utama yaitu cara agen menemukan destinasi dan cara agen menuju destinasi.

3.2.1 *Seeking*

Bagian ini adalah bagian dimana agen pada simulasi menentukan titik tujuan pergerakannya(destinasi). Agen *leader* dan *follower* memiliki jenis tujuan yang berbeda. Sebagaimana ditunjukkan pada *flowchart* di gambar 3.4 agen *leader* memiliki titik tujuan yang telah ditentukan sedangkan *follower* memiliki titik tujuan pada *leader* tiap *follower*. Pada salah satu skenario pengujian, agen *leader*



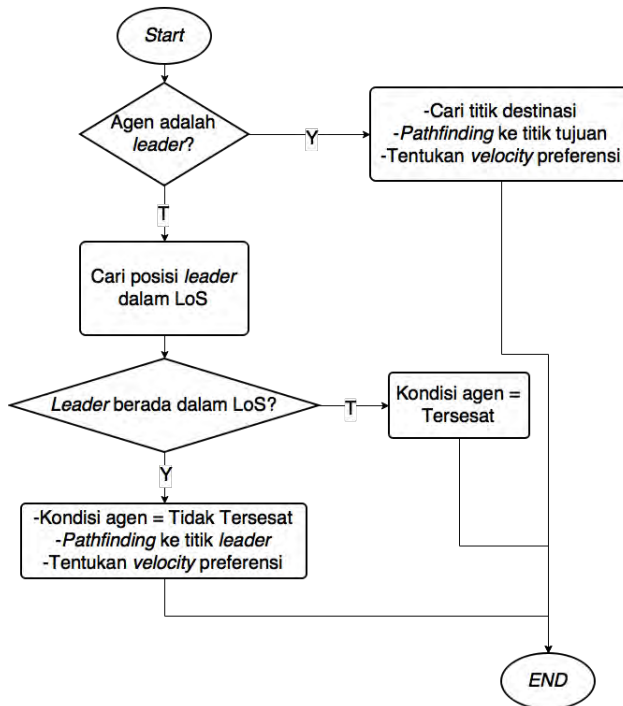
Gambar 3.3: *Flowchart* alur pergerakan agen setiap *cycle*

memiliki beberapa titik tujuan dengan *waypoint* tertentu. Agen *leader* akan mencari jalur menuju destinasinya. Pencarian jalur dari titik agen menuju titik tujuan memakai algoritma A*.

Untuk agen *follower*, setiap agen memiliki seorang *leader* yang telah ditentukan dan posisi *leader*. Agen *follower* pertama kali akan

mencari *leader* dalam LoS(*Line of Sight*). Bila pada siklus tersebut, agen tidak dapat menemukan *leader*, maka status agen tersebut dianggap tersesat.

Agen *follower* yang telah menemukan *leader*-nya di dalam LoS akan dianggap sebagai agen yang tidak tersesat. Selanjutnya agen *follower* akan mencari jalur menuju posisi *leader* dengan algoritma A*. Hasil pencarian jalur ketitik tujuan masing-masing agen akan menghasilkan parameter jarak dan arah yang akan disimpan untuk proses selanjutnya dalam bentuk *velocity* preferensi.



Gambar 3.4: Flowchart seeking

3.2.2 *Steering*

Setelah agen menentukan titik tujuannya serta mendapatkan *velocity preferensi* dari proses *seeking* pada subbab 3.2.1 maka proses selanjutnya adalah pengarahan agen menuju destinasinya yaitu *steering*. Dapat dilihat pada *flowchart* 3.5, agen *follower* bila tidak tersesat akan mengambil parameter *velocity* preferensi dari proses *seeking*. Parameter yang dipakai yaitu jarak dari agen tersebut ke *leader*-nya. Dari parameter ini akan ditentukan kelajuan (*speed*) agen dimana semakin jauh jaraknya akan semakin cepat kelajuannya agar bisa mengejar *leader*.

Dapat dilihat pada persamaan 3.7, kelajuan agen *follower* untuk *velocity* referensi bergantung pada kelajuan agen *leader* dan pengali kecepatan. Kelajuan pengejaran sendiri diberikan pada persamaan 3.8. Kelajuan pengejaran bertambah secara eksponensial berbanding lurus dengan penambahan jarak dari agen *follower* ke *leader*. Hal ini berarti semakin jauh agen *follower* dari *leader*, maka penambahan kelajuan akan bertambah tinggi agar bisa mengimbangi ketertinggalan dengan agen *leader*. Persamaan kelajuan agen *follower* secara keseluruhan dapat dilihat di persamaan 3.9.

$$V_{follower} = V_{leader} \times SM \quad (3.7)$$

$$SM = \exp\left(\frac{D_{agent} - D_{eq}}{D_{LoS}}\right)^n \quad (3.8)$$

$$V_{follower} = V_{leader} \times \exp\left(\frac{D_{agen} - D_{eq}}{D_{LoS}}\right)^n \quad (3.9)$$

$V_{follower}$	= Kelajuan agen <i>follower</i> .
V_{leader}	= Kelajuan agen <i>leader</i> .
SM	= Pengali kecepatan.
D_{agent}	= Jarak agen <i>follower</i> terhadap posisi <i>leader</i> .
D_{eq}	= Jarak yang diinginkan untuk agen <i>follower</i> memiliki kelajuan yang sama dengan agen <i>leader</i> .
D_{LoS}	= LoS agen <i>follower</i> .
n	= Konstanta, semakin rendah nilainya akan membuat penambahan nilai mendekati linier.

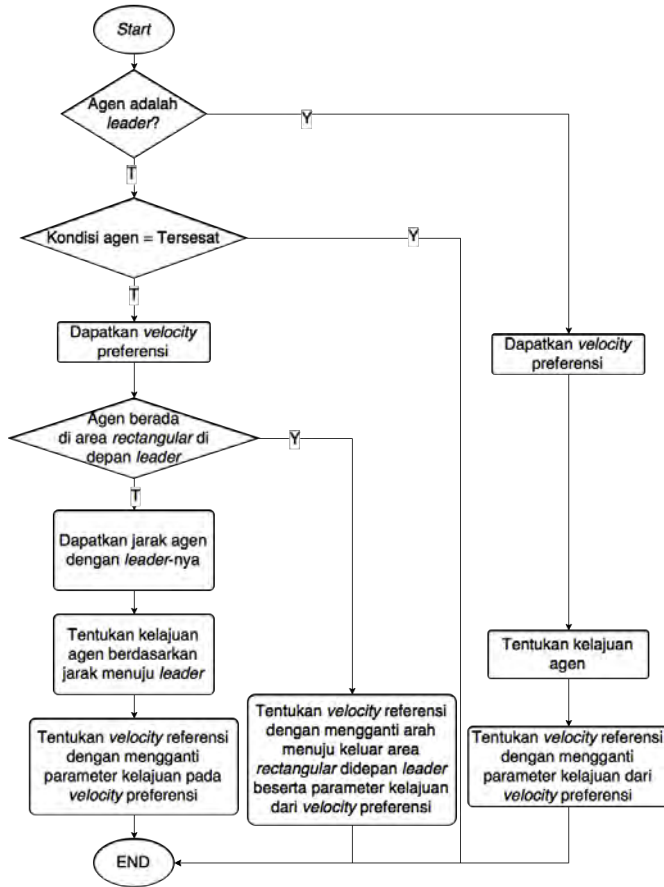
Untuk agen *leader*, parameter kelajuan yang diubah dari *velocity* preferensi ditentukan secara konstan. Hal ini mengakibatkan kelajuan pada *velocity* referensi *leader* adalah *velocity* ideal. Sehingga perubahan *velocity* hanya tergantung dari proses selanjutnya. Kelajuan agen yang dipilih akan mengubah parameter kelajuan yang berada *velocity* preferensi. *Velocity* ini disimpan sebagai *velocity* referensi.

3.2.3 Avoiding Obstacle

Avoiding Obstacle adalah penghindaran tabrakan baik itu rintangan statis maupun agen yang bergerak. Pengindaran dilakukan dengan menentukan gerak agen yang valid melalui pemilihan *valid velocity* berdasarkan algoritma RVO. Keluaran dari proses ini adalah penentuan *velocity* agen yang akan dipakai untuk siklus gerak saat itu.

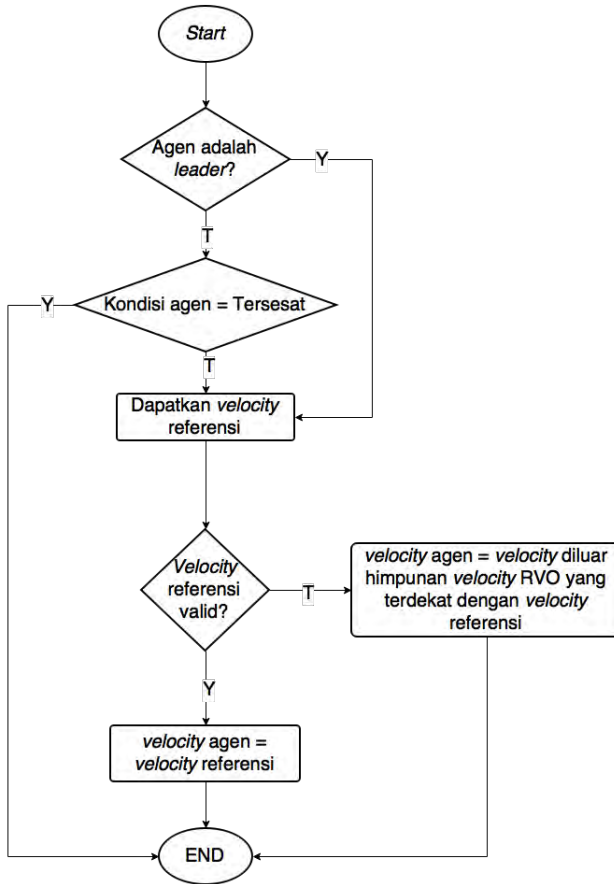
Alur proses yang ditunjukkan pada *flowchart* di gambar 3.6 menggambarkan alur penentuan *velocity* agen. Untuk seluruh agen baik itu *leader* maupun *follower* (yang tidak tersesat), *Velocity* referensi dari proses 3.2.2 akan diambil untuk pengecekan validitas dari *velocity* referensi tersebut. *Velocity* referensi valid bila *velocity* tersebut tidak berada dalam *collision cone* dari RVO yang artinya, *velocity* referensi tidak mengakibatkan *collision*.

Bila *velocity* referensi dinyatakan valid maka *velocity* referensi telah sah menjadi *velocity* agen untuk siklus saat itu. Sebaliknya bila *velocity* referensi tersebut tidak valid yang artinya *velocity* tersebut akan mengakibatkan tabrakan, maka sebagaimana pada penjelasan RVO dalam penghindaran *collision*[1], *velocity* agen akan



Gambar 3.5: Flowchart steering

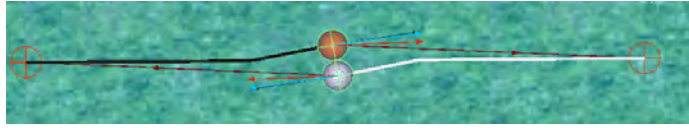
dipilih dari *velocity* diluar himpunan *velocity* RVO yang terdekat dengan *velocity* referensi. Dalam simulasi ini terdapat tambahan *invalid velocity* yaitu *invalid velocity* untuk separasi antar agen. Seperti yang terlihat pada Gambar 3.7, separasi antar agen mengakibatkan *velocity* yang berada di area separasi menjadi *invalid*.



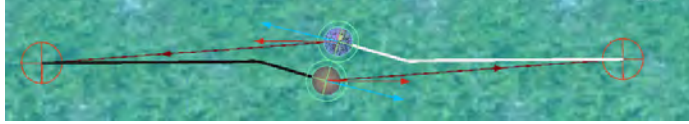
Gambar 3.6: *Flowchart collision avoidance*

3.2.4 Detecting collision

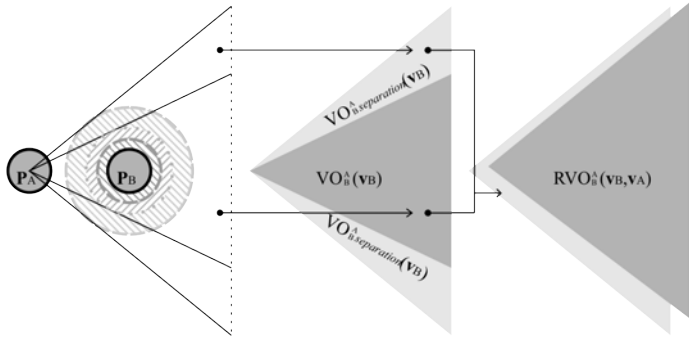
Tahap terakhir setelah gerakan agen ditentukan adalah *detecting collision* untuk mendeteksi terjadinya tabrakan. Tabrakan ini dideteksi dengan menggunakan *collider* yang ada pada unity. *Collider* yang dipakai adalah *capsule collider* dan bekerja sebagaimana algoritma deteksi tabrakan *circle to circle detection* yang dapat dilihat di flowchart pada gambar 3.8. *Collider* ini akan mendeteksi



(a) RVO sebelum penambahan parameter separasi



(b) RVO dengan penambahan parameter separasi



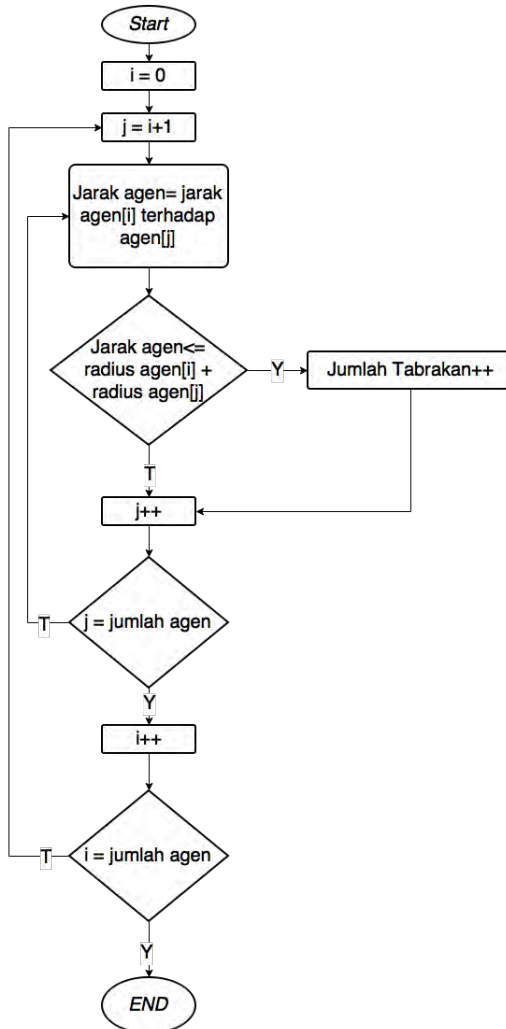
(c) Penambahan parameter separasi pada RVO

Gambar 3.7: Desain RVO untuk simulasi

tabrakan bila *collider* yang satu dengan yang lainnya terjadi kontak fisik.

3.3 Alur Implementasi Sistem

Pengimplementasian sistem simulasi dibagi menjadi tiga tahapan, tahap tersebut meliputi lingkungan simulasi, agen simulasi, dan pengambilan data. Ketiganya diimplementasikan pada *game engine* Unity.



Gambar 3.8: Flowchart collision detection

3.3.1 Pembuatan Lingkungan Simulasi

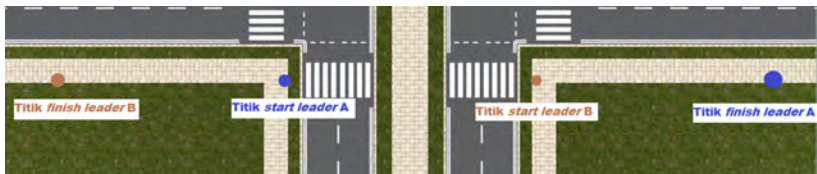
Lingkungan simulasi kerumunan yang dibuat disesuaikan dengan skenario simulasi. Secara garis besar objek lingkungan simula-

si kerumunan dihimpun menjadi dua jenis berdasarkan kemampuan agen bernavigasi yaitu *walkable* dan *unwalkable*. Sehingga perlu ada pendefinisian pada lingkungan simulasi bagian mana yang dapat dilalui agen dan tidak dapat dilalui untuk masing-masing scenario. Penskalaan ukuran yang dipakai dalam simulasi ini sebesar 2:1 yang berarti dua unit ukuran virtual di Unity sama dengan 1 meter ukuran panjang di dunia nyata.

Seluruh lingkungan dibuat dengan kondisi yang sama pada semua skenario pengujian. Seluruh area navigasi lingkungan tidak memiliki variasi friksi yang dapat memengaruhi kelajuan agen. Tidak ada *environmental hazard* seperti hujan, badai dan sebagainya. Dengan begitu pengaruh lingkungan terhadap navigasi agen hanya pada pengaruh rintangan statis seperti pada skenario *narrow passage* dan *tour guide*.

1. *Crossroad Scenario*

Pada simulasi ini, lingkungan simulasi dibentuk dengan area jalan agen berupa trotoar, dan penyebrangan jalan. Area yang ditunjukkan pada gambar 3.10 adalah area yang dapat dilewati (*walkable*) agen dan dibatasi sebesar 220x16 unit. Di area tersebut tidak terdapat rintangan dari lingkungan yang berarti penghindaran agen hanya terhadap agen lain.

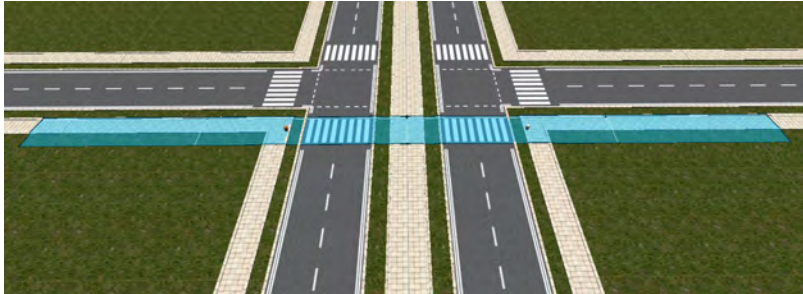


Gambar 3.9: Desain lingkungan simulasi *crossroad scenario*

Penempatan titik awal dan titik akhir masing-masing *leader* ditunjukkan pada gambar 3.9. Koordinat *leader* dapat dilihat pada tabel 3.1. Dengan kondisi koordinat *leader* yang didesain tersebut, kedua kelompok kerumunan akan bertemu ditengah penyebrangan yang merupakan tantangan pada skenario ini.

Tabel 3.1: Titik awal dan akhir agen *leader* dalam *crossroad scenario*

<i>Leader</i>	Titik awal	Titik akhir
A	35;5	-98;5
B	-35;5	98;5



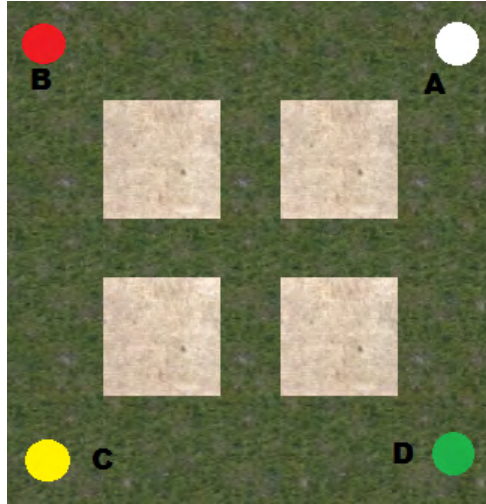
Gambar 3.10: Area *walkable* (biru) untuk *crossroad scenario*

2. *Narrow Passage* Lingkungan simulasi untuk skenario ini dibentuk pada area seluas 85x85 unit. Gambar 3.12 menunjukkan lingkungan simulasi yang telah selesai beserta area *walkable* agen. Sebagaimana ditunjukkan pada gambar tersebut, terdapat 4 buah rintangan berupa empat buah kotak dengan ukuran 10x10 yang ditempatkan sedemikian rupa sehingga terbentuk jalan sempit selebar 5 unit antar rintangan tersebut yang diistilahkan dengan *narrow-passage* yang mana merupakan tantangan pada skenario ini. Pada skenario ini

Tabel 3.2: Koordinat titik yang ditunjukkan gambar 3.11

<i>Leader</i>	Titik awal
A	50;50
B	15;50
C	15;15
D	50;15

terdapat agen leader akan ditempatkan pada simulasi seperti yang ditunjukkan pada gambar 3.11 dimana titik awal agen

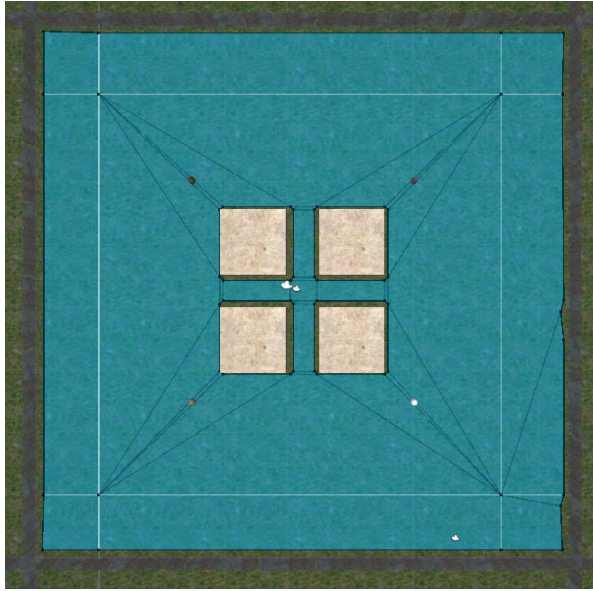


Gambar 3.11: Desain lingkungan simulasi *narrow-passage scenario*

leader A berada pada titik A, agen *leader* B berada pada titik B, agen *leader* C berada pada titik C, agen *leader* D berada pada titik D. Adapun titik tujuan dari masing-masing agen *leader* adalah titik awal agen *leader* yang bersebrangan. Sehingga titik akhir agen *leader* A adalah titik awal agen *leader* C. Titik akhir agen *leader* B adalah titik awal agen *leader* D. Titik akhir agen *leader* C adalah titik awal agen *leader* A. Titik akhir agen *leader* D adalah titik awal agen *leader* B. Keterangan koordinat titik yang ditunjukkan pada gambar 3.11 dapat dilihat pada tabel 3.2.

Dengan posisi penempatan agen *leader* yang demikian, seluruh kerumunan akan terkondisikan untuk melewati area *narrow-passage* yang terbentuk oleh rintangan kotak. Hal ini mengakibatkan terjadinya kepadatan agen yang tinggi di titik pertemuan keempat kerumunan di dalam *narrow passage*.

3. *Tour Guide Scenario* Skenario ini merupakan skenario yang dibangun dengan mengambil model lingkungan dari dunia nyata, dengan maksud untuk menguji sistem navigasi multiagen



Gambar 3.12: Area *walkable* untuk *narrow-passage scenario*

pada kasus kerumunan tur. Skenario ini juga sebagai contoh implementasi navigasi multiagen dengan RVO dengan relasi *leader-follower*. Pada skenario ini, agen *leader* disini adalah *tour guide* dan agen *follower* adalah kerumunan pengunjung.

Sebagaimana telah dijelaskan pada bagian 3.1.1, lingkungan simulasi dibentuk berdasarkan *layout* pameran alutsista TNI yang ditunjukkan pada gambar 3.13. Hasil pembuatan ulang dalam Unity dapat dilihat pada gambar 3.14. Properti pada lingkungan simulasi dibuat seakurat mungkin dengan model 3D yang didapatkan secara gratis dari berbagai sumber. Sebagian peralatan yang dipamerkan di dunia nyata diganti dengan objek lain yang masih merepresentasikan peralatan yang dipamerkan karena keterbatasan ketersediaan model 3D. Posisi penempatan diusahakan persis dengan penempatan properti



Gambar 3.13: Denah pameran

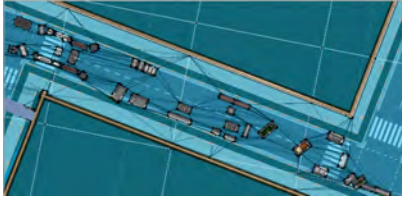
didunia nyata. Area *walkable* dapat dilihat di gambar 3.15.



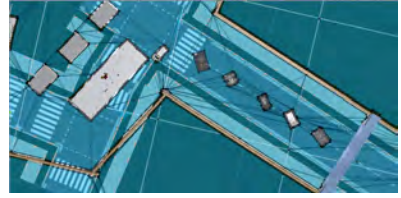
Gambar 3.14: Layout skenario tour guide

Properti pameran pada skenario ini dibuat sebagai halangan *unwalkable*. Dengan kata lain properti pameran pada skenario ini dianggap sebagai rintangan yang perlu dihindari agen pada skenario ini. Disinilah letak tantangan *tour guide scenario* dimana banyak rintangan di jalur perjalanan agen.

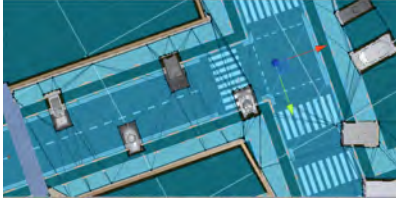
Berbeda dengan skenario sebelumnya, titik tujuan satu agen *leader* lebih dari satu dengan sistem *waypoint*. Terdapat de-



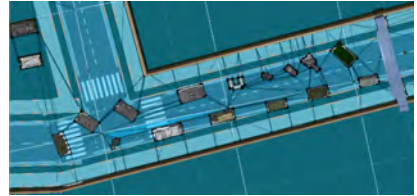
(a) Area *walkable* bagian utara



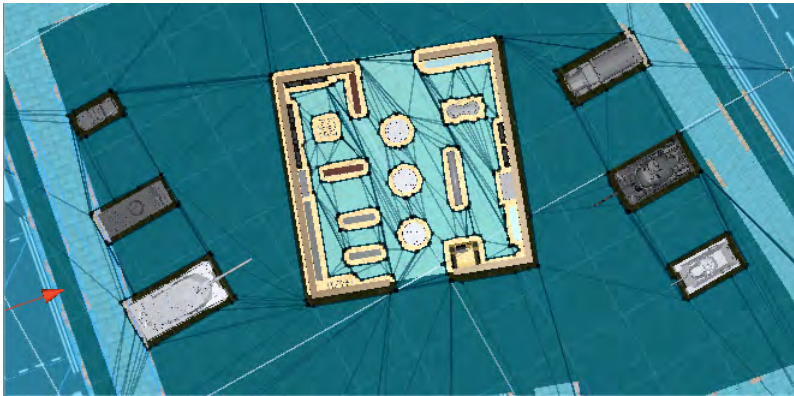
(b) Area *walkable* bagian selatan



(c) Area *walkable* bagian barat



(d) Area *walkable* bagian timur



(e) Area *walkable* bagian tengah

Gambar 3.15: area *walkable* pada *tourguide scenario*

lapan *waypoint* untuk setiap *leader*. Titik mulai dan *waypoint* masing-masing *leader* dapat dilihat di tabel 3.3

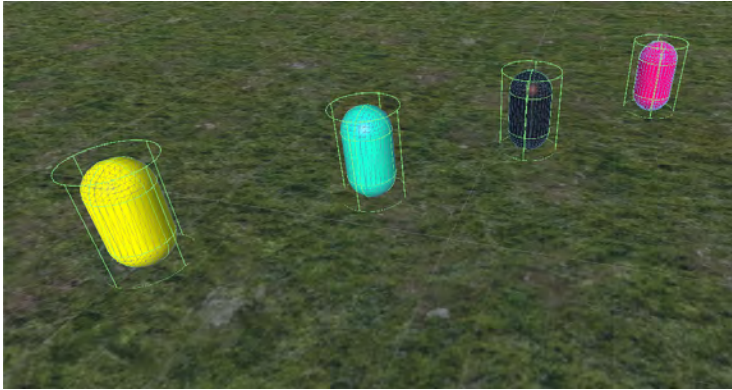
Tabel 3.3: Titik mulai dan *waypoint* agen *leader tour guide scenario*

Waypoint	Leader A (x;y)	Leader B(x;y)	Leader C(x;y)	Leader D(x;y)
0 (Titik awal)	150; -58	51; 205	-43.5; -158	-268.57; -59
1	52.25; -12.44	52.34; -19.4	-26.36; -63.13	-35.7; -62.95
2	51.77; 188.66	139.63; -48.83	-263.37; - 20.2	-34.2; -141.5
3	-24.23; 8.42	53.21; -43.84	-12; 5	47.27; -20.84
4	-261.9; -22.27	-32.9; -141.45	51.76; 191.83	136.93; -60.61
5	-34.93; -62.93	-37.13; -60.29	53.06; -20.25	54.2; 1.74
6	-34.13; -139.47	-265.47; - 20.74	139.41; -53.67	56.88; 196.49
7	21; 0	-12.5; 5	-18.43; -63.59	25; -2.5
8	200; -75	44; 323.5	-54; -220	-280; -125

3.3.2 Pembuatan Agen Simulasi

Agen dalam simulasi kerumunan ini dianggap sebagai objek *capsule* yang bertranslasi pada bidang dimensi dua. Desain agen dapat dilihat pada gambar 3.16. Pada pengujian standar dimana ukuran agen tidak bervariasi, diameter agen dibuat sebesar 1 unit. Agen ini didefinisikan sebagai *navmesh agent* pada unity agar agen ini mampu bernavigasi pada *navigation mesh* yang *walkable* di lingkungan simulasi yang telah dijelaskan pada bagian 3.1.1. Dimana *navmesh agent* ini merupakan salah satu properti dari RVO untuk mendefinisikan *valid velocity* berdasarkan lebar *navmesh agent*.

Pada pengaturan *navmesh agent* pada Unity, terdapat tambahan *velocity* yang *invalid* selain yang disebabkan karena *velocity* yang dapat mengakibatkan *collision*. Hal ini dikarenakan adanya separasi seperti yang dijelaskan pada bagian 3.2.2, jarak separasi l_s



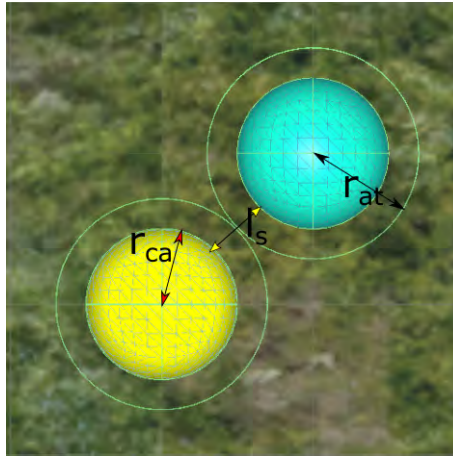
Gambar 3.16: Agen simulasi kerumunan

yang ditunjukkan di gambar 3.17 harus mengakibatkan *invalid velocity* pada *velocity* yang menghasilkan gerakan agen didalam area separasi. Sehingga *navmesh agent* memiliki radius r_{at} sebesar radius agen (r_{ca}) ditambah dengan separasi agen l_s dibagi dua (karena separasi antar agen diberi beban yang sama). Dengan begitu navigasi agen yang dihasilkan RVO sudah termasuk separasi antar agen sebagaimana ditunjukkan pada gambar 3.7.

Karena tidak ada variasi jenis agen, maka seluruh agen diasumsikan memiliki kondisi sebagaimana manusia dewasa normal. Sebagaimana dijelaskan pada subbab 3.2.2, Kelajuan referensi agen *leader* diatur stabil sebesar 1,2 meter/detik. Kelajuan ini merupakan kelajuan umum untuk para pejalan kaki[11]. Adapun kelajuan referensi agen *follower* bergantung terhadap jarak agen tersebut terhadap *leader*-nya sebagaimana pada persamaan 3.9. Kelajuan terpilih agen masih dapat berubah menyesuaikan dengan tindakan penghindaran dalam kerumunannya sesuai dengan algoritma RVO sebagaimana telah dijelaskan pada subbab 3.2.3.

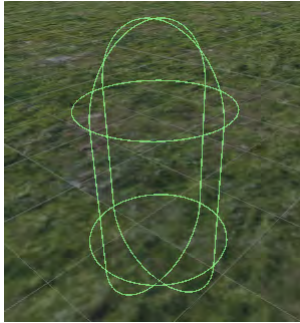
Pada seluruh skenario simulasi, agen *leader* dianggap sebagai agen yang telah mengenal area lingkungan skenario sehingga agen ini diasumsikan telah mengetahui kemana dia akan menuju. Untuk agen *follower* sendiri, agen ini dianggap tidak mengetahui area skenario sehingga harus mengikuti *leader*-nya dalam mencapai tu-

juan kelompoknya. Hal ini meniru kondisi seperti pada pada tur di dunia nyata, dimana pengunjung tur(*follower*) diasumsikan belum pernah berkunjung pada area tur. Adapun pemandu tur(*leader*) telah mengetahui dengan pasti kemana harus mengarahkan kerumunan tur ini. Agen *follower* di semua skenario pengujian memiliki *LoS* sebesar 50 meter.

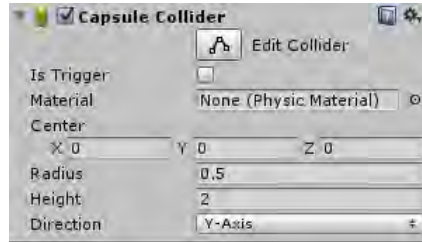


Gambar 3.17: Lebar agen dan separasi antar agen

Untuk keperluan deteksi tabrakan, maka perlu ditanam *collider* yang merupakan sistem dalam Unity untuk *physics engine* dalam mendeteksi tabrakan. *Collider* yang dipakai mengikuti bentuk agen yaitu *capsule collider* dengan ukuran yang juga menyesuaikan ukuran agen. Gambar 3.18 adalah gambar *capsule collider* pada unity beserta propertinya. *Collider* akan mendeteksi tabrakan bila jarak antar titik pusat *collider* lebih rendah dari jumlah radiusnya.



(a) Unity capsule collider



(b) Properties capsule collider

Gambar 3.18: Collider yang ditanam pada tubuh agen

BAB 4

PENGUJIAN DAN ANALISA

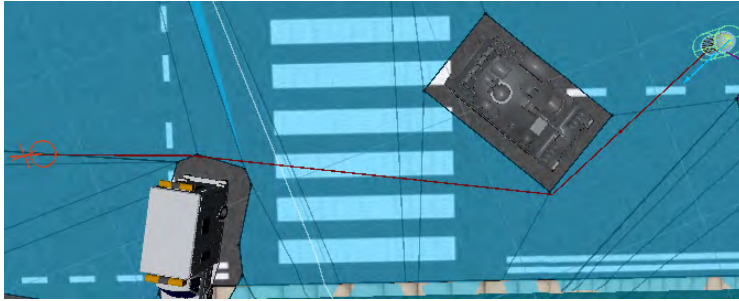
Pada bab ini akan dipaparkan hasil pengujian dan analisa dari desain sistem dan implementasi yang telah dibahas di bab 3. Pengujian akan ditujukan untuk memperlihatkan apakah sistem simulasi telah berjalan dengan baik serta implementasi RVO dengan relasi *leader & follower* telah memberikan hasil yang optimal. Untuk dapat memperlihatkan hasil keandalan RVO akan dilakukan pengukuran dengan memperlihatkan hasil grafik dari berbagai macam data yang diekstraksi sebagaimana disebutkan pada bagian 3.1.2. Pengujian dilakukan menggunakan editor pada perangkat lunak Unity versi 5.2.0f *personal edition* pada sistem operasi Windows 7.

4.1 Pengujian Sistem Simulasi

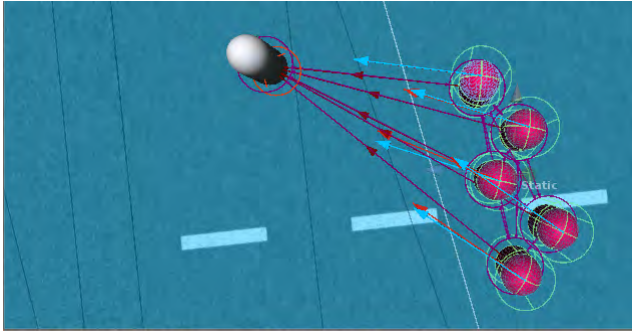
Pada bagian ini pengujian dilakukan untuk memastikan apakah simulasi kerumunan dapat berjalan atau tidak, baik dari pergerakan agen hingga ekstraksi data. Proses yang akan diuji dalam penelitian ini diantaranya *seeking*, *steering*, *collision avoidance* dan *collision detection* yang akan dijabarkan pada bagian selanjutnya. Kemudian dilanjutkan dengan pengujian ekstraksi data.

4.1.1 Pengujian *Seeking*

Proses pertama pada penentuan pergerakan agen disetiap siklusnya adalah *seeking* sebagaimana disebutkan pada 3.2.1. Pada pengujian ini akan diperlihatkan kemampuan agen dalam mencari titik destinasinya. Pada gambar 4.1 diperlihatkan hasil dari proses *seeking* dimana 4.1(a) merupakan hasil *seeking* oleh agen *leader* 4.1(b) oleh agen *follower*. Agen *leader* dapat menemukan titik destinasinya yaitu rute yang telah ditentukan. Agen *follower* menemukan titik destinasinya yaitu agen *leader* yang harus dia ikuti. Jalur yang didapat ditunjukkan oleh garis merah. Jalur tersebut merupakan hasil algoritma A*. Hasil ini disimpan dalam bentuk *velocity preferensi* yang berisi arah serta jarak ke titik tujuan.



(a) Agen *leader*(putih)



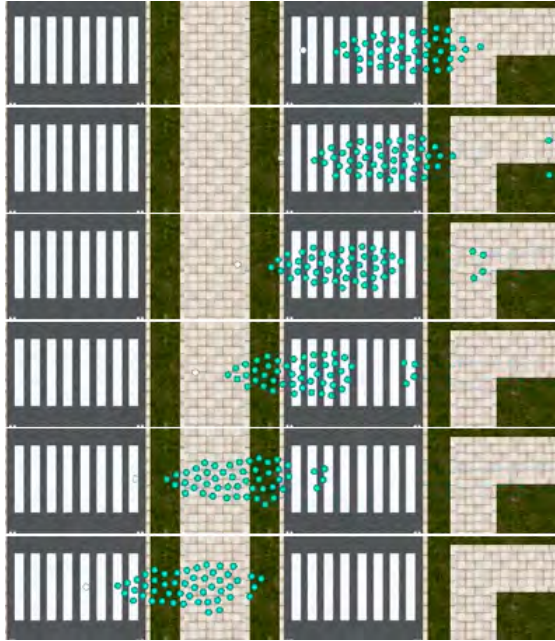
(b) Agen *follower* (merah)

Gambar 4.1: Hasil *seeking* agen *leader* & *follower*

4.1.2 Pengujian *Steering*

Setelah agen menentukan titik tujuannya serta mendapatkan *velocity preferensi* dari proses *seeking* maka proses selanjutnya yang diujikan adalah pengarahan agen menuju destinasiya yaitu *steering*. Pengujian pertama adalah pengecekan pemilihan kelajuan agen dalam pengejaran terhadap leader *leader*. Dapat dilihat pada gambar 4.2, terdapat empat agen *follower* yang tertinggal dari kelompoknya sehingga agen tersebut menambah kelajuan untuk mengejar ketertinggalannya.

Pengujian selanjutnya untuk tahapan *steering* adalah pengujian pada pengarahan agen *follower* untuk memberikan jalan kepada

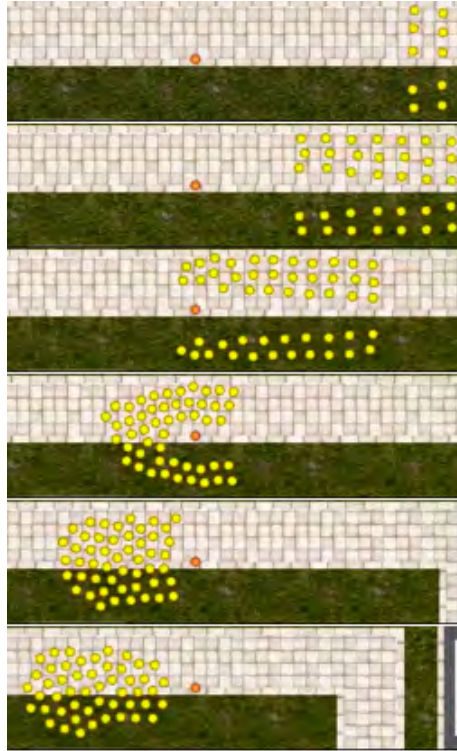


Gambar 4.2: Pengejaran oleh agen *follower* yang tertinggal

leader-nya bila agen *follower* berada di depan *leader*. Hasil pengujian ditunjukkan pada gambar 4.3 dimana agen *follower* diarahkan keluar dari area *rectangular* yang berada didepan *leader*. Parameter *velocity* hasil dari *steering* ini disimpan sebagai *velocity* referensi.

4.1.3 Pengujian *Collision Avoidance*

Pada tahap pengujian ini, agen simulasi yang telah mendapatkan *velocity* referensinya akan melakukan proses pengecekan validitas dari *velocity* referensi tersebut dengan algoritma RVO. Pengujian dilakukan dengan menggerakkan dua agen simulasi yang memiliki arah gerak berlawanan sehingga akan mengakibatkan tabrakan bila kedua agen tersebut bergerak dengan tetap menggunakan *velocity* referensi. Proses navigasi penghindaran agen dengan RVO akan mengubah *velocity* agen diluar *velocity obstacle* yang



Gambar 4.3: *Steering* agen *follower* yang berada di depan *leader*-nya

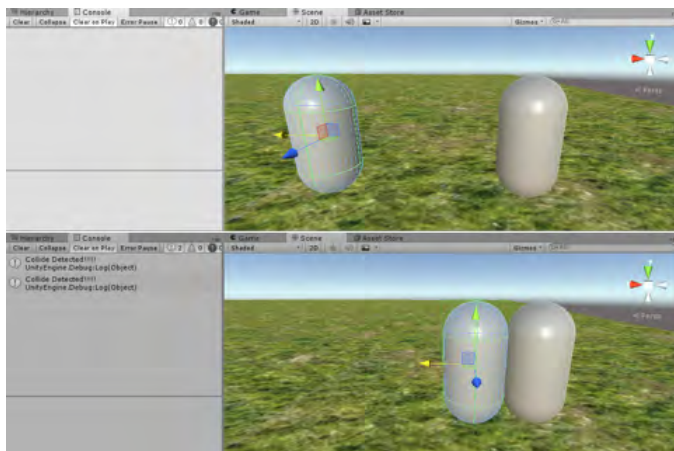
mengakibatkan tabrakan. Hasil dari pengujian ini dapat dilihat pada gambar 4.4.

4.1.4 Pengujian *Collision Detection*

Collision detection merupakan proses terakhir pada satu siklus kalkulasi pergerakan agen. Pada proses ini akan dilakukan pengujian deteksi tabrakan oleh *collider* yang ditanam pada tubuh agen. Pada gambar 4.5 kedua agen yang awalnya tidak bertabrakan di translasikan sehingga menabrak satu sama lain. Ketika terjadi tabrakan, akan muncul *log* peringatan "*collide detected!!!!*" yang berarti kedua agen mendeteksi terjadinya tabrakan.



Gambar 4.4: Hasil penghindaran tabrakan dengan menggunakan RVO



Gambar 4.5: Hasil pengujian deteksi tabrakan

4.1.5 Pengujian ekstraksi data

Pengujian ini dimaksudkan untuk memeriksa sistem pengambilan data dan memastikannya telah berjalan sesuai yang diinginkan. Pengujian ini dilakukan dengan mencoba seluruh fungsi dalam ekstraksi data dan memastikan apakah telah berjalan dengan baik atau tidak. Hasil dari pengujian seluruh fungsi berjalan dengan sesuai seperti pada desain sistem yang dijelaskan pada bagian 3.1.2. Data yang di ekstraksi ditampilkan pada *user interface* simulasi seperti pada gambar 3.2. Hasil yang didapatkan dari pengujian kesesuaian sistem permainan ini terdapat pada Tabel 4.1.

Tabel 4.1: Hasil pengujian fungsi pengambilan data

No.	Nama data	Fungsi dapat berjalan
1	Jumlah <i>frame</i>	Ya
2	Jumlah <i>tabrakan</i>	Ya
3	Waktu simulasi	Ya
4	<i>Collision per frame</i>	Ya
5	Rata-rata <i>frame rate</i>	Ya
6	Jumlah agen tersesat	Ya

4.2 Hasil Simulasi

Pada bagian ini akan ditampilkan data yang didapat dari simulasi kerumunan pada ketiga skenario simulasi yang diujikan (*cross-road*, *narrow passage*, *tour guide*). Simulasi dijalankan pada komputer dengan spesifikasi prosesor Intel Core i5-3470 3.20 GHZ dengan RAM sebesar 8 GB serta GPU Radeon HD7700 VRAM 1GB. Setiap skenario simulasi dijalankan dengan jumlah agen bervariasi hingga maksimal 200 agen. Setiap skenario dijalankan dengan memakai relasi *leader & follower* dan tanpa ada relasi antar agen. Dari kedua hasil ini didapatkan data keefektifan RVO dengan memakai relasi *leader & follower* dan tanpa relasi *leader & follower*.

Secara umum pada setiap skenario simulasi, akan terjadi tabrakan pada tingkat keramaian tertentu. Tabrakan ini diakibatkan karena tingkat kerumunan yang tinggi mengakibatkan kalkulasi

perhitungan yang lebih kompleks. Hal ini terjadi akibat semakin banyaknya agen yang harus ditentukan pergerakannya secara lokal tanpa ada koordinasi. Selain itu pada penggunaan relasi *leader & follower*, gerak agen *follower* semakin terikat pada gerakan *leader*-nya. Perhitungan gerak agen dilakukan pada tiap *frame*. Sehingga bila pada suatu *frame* telah ditentukan *velocity* seluruh agen, maka perubahan *velocity* selanjutnya harus menunggu *frame* selanjutnya. Bila sebelum *frame* selanjutnya tercapai, *velocity* agen berada dalam *collision cone*, agen bisa bertabrakan. Hal ini diakibatkan belum ada perubahan pada *velocity* agen dikarenakan *frame* selanjutnya belum tercapai.

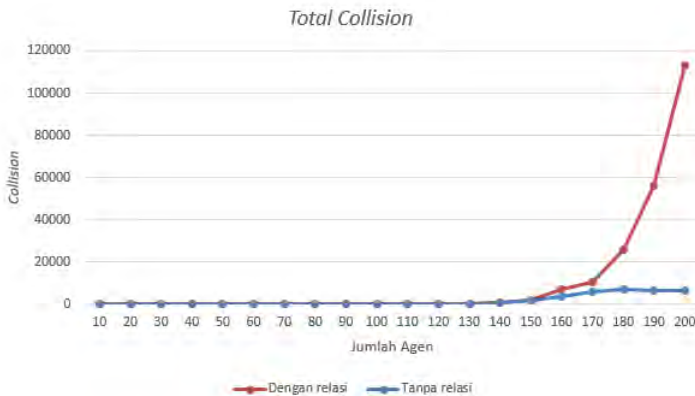
4.2.1 *CrossRoad Scenario*

Simulasi *crossroad scenario* mempertemukan dua kelompok kerumunan dengan arah tujuan berlawanan. Pertemuan kelompok agen ini bisa mengakibatkan tabrakan pada suatu tingkat keramaian kerumunan. Dari data yang didapat, terjadi peningkatan tabrakan secara eksponensial pada simulasi dengan relasi *leader & follower* yang dapat dilihat di grafik pada gambar 4.6. Hal ini terjadi karena penghindaran agen *follower* kedua kelompok terikat pada pengejarannya terhadap *leader* tiap kelompok. Hal ini berakibat penumpukan kerumunan karena keterbatasan lebar jalan yang bisa dilalui, dimana pada jumlah agen diatas 150 agen terjadi *overlapping* antar agen karena kerumunan telah sangat berdesakan.

Hasil simulasi *crossroad scenario* tanpa relasi menunjukkan persentasi kenaikan jumlah tabrakan yang lebih sedikit sebagaimana ditunjukkan pada grafik 4.11. Hal ini dikarenakan tidak adanya keterikatan tiap-tiap agen terhadap agen *leader*. Namun waktu simulasi yang dibutuhkan lebih lama dibandingkan waktu yang diperlukan kerumunan dengan *leader* sebagaimana ditunjukkan pada grafik di gambar 4.9. Secara umum, persentase penurunan waktu simulasi pada grafik 4.12 dengan *leader-follower* terhadap skenario tanpa relasi bernilai positif dengan rata-rata sebesar 18%. Data tersebut menunjukkan penggunaan RVO dengan relasi *leader-follower* mereduksi waktu yang diperlukan kelompok agen untuk menuju tujuannya. Grafik pada gambar 4.7 menunjukkan jumlah *frame* selama simulasi berjalan. Penambahan waktu simulasi juga berakibat

pada jumlah *frame* yang meningkat sejalan dengan penambahan waktu.

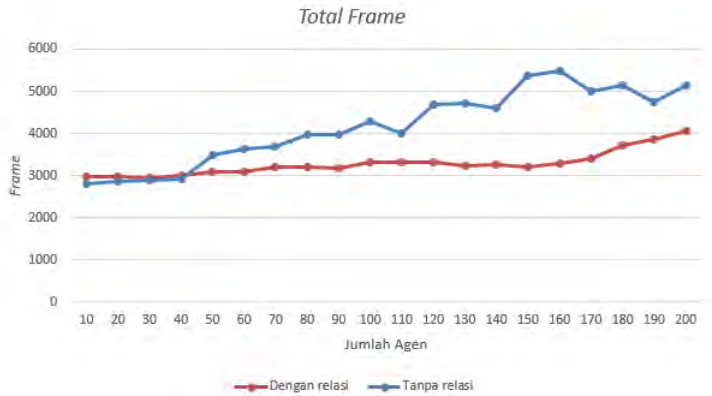
Rata-rata *frame rate* di gambar 4.10 relatif stabil pada nilai rata-rata 60,14 FPS hingga jumlah agen diatas 170 agen dengan relasi *leader-follower*. Pada jumlah agen tersebut telah terjadi penurunan *frame rate* hingga mencapai 54,34 FPS akibat terjadinya tabrakan yang cukup masif, bahkan mengakibatkan nilai CpF yang ditunjukkan grafik CpF di gambar 4.8 berada diatas lima CpF. Bila dibandingkan dengan hasil skenario yang lain, nilai CpF diatas satu CpF hanya terjadi pada skenario ini.



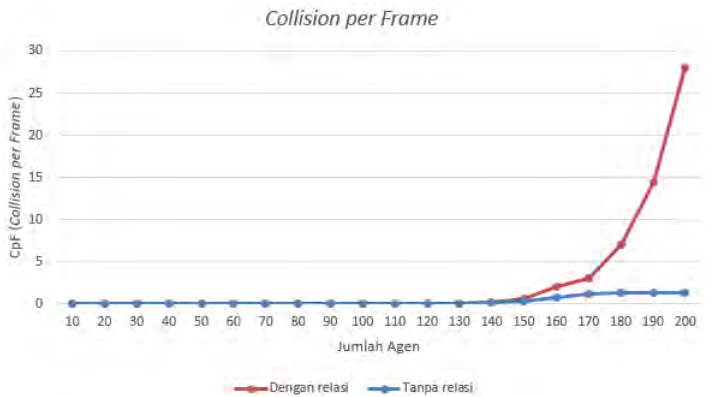
Gambar 4.6: Grafik jumlah tabrakan yang terjadi pada simulasi *cross-road scenario*

4.2.2 *Narrow Passage Scenario*

Pada skenario *narrow passage*, empat kelompok di posisi pojok *environment* saling bertukar posisi. Ditengah jalur pertukaran posisi ini akan ada jalan sempit terbentuk dari empat buah rintangan berbentuk kotak. Tabrakan yang terjadi pada simulasi ini terjadi dimulai pada jumlah agen sebanyak 104 sebagaimana ditunjukkan pada grafik di gambar 4.13. Pada grafik presentase kenaikan



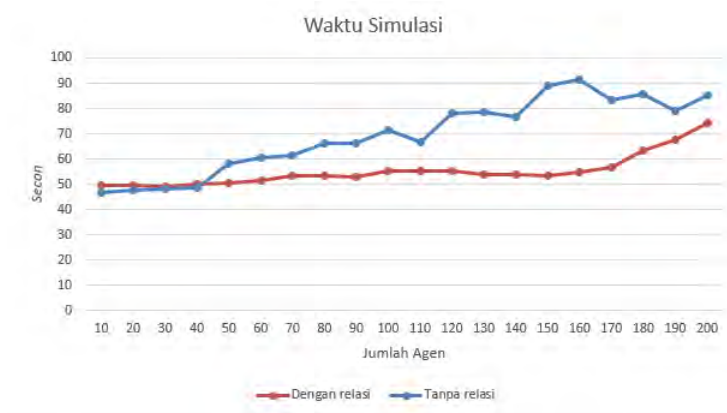
Gambar 4.7: Grafik jumlah frame pada simulasi *crossroad scenario*



Gambar 4.8: Grafik *collision per frame* untuk simulasi *crossroad scenario*

tabrakan tiap agen di gambar 4.18, kenaikan tabrakan pada agen dengan relasi *leader-follower* lebih tinggi dari agen tanpa relasi.

Pada simulasi tanpa relasi, sebagian agen memilih jalan memutar rintangan (tidak melewati *narrow passage*). Hal ini terjadi pada jumlah agen 104 dan di atasnya. Penyebabnya dikarenakan ke-



Gambar 4.9: Grafik waktu simulasi *crossroad scenario*



Gambar 4.10: Grafik rata-rata *frame rate* untuk simulasi *crossroad scenario*

ramaian pada *narrow passage* menyebabkan kepadatan yang tinggi. Tingkat kepadatan yang terjadi mengakibatkan sebagian agen yang berada di luar *narrow passage* cenderung memutar rintangan ketika terjadi hambatan di *narrow passage*. Pada simulasi dengan relasi *leader & follower*, agen *follower* berkerumun di *narrow passa-*



Gambar 4.11: Grafik peningkatan tabrakan agen untuk simulasi *cross-road scenario*

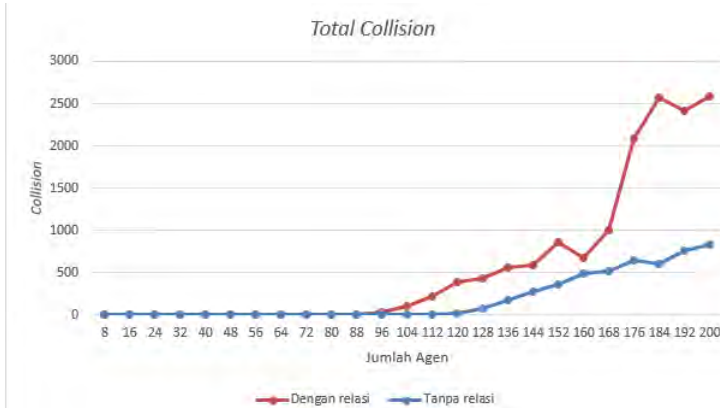


Gambar 4.12: Grafik penurunan waktu simulasi *crossroad scenario*

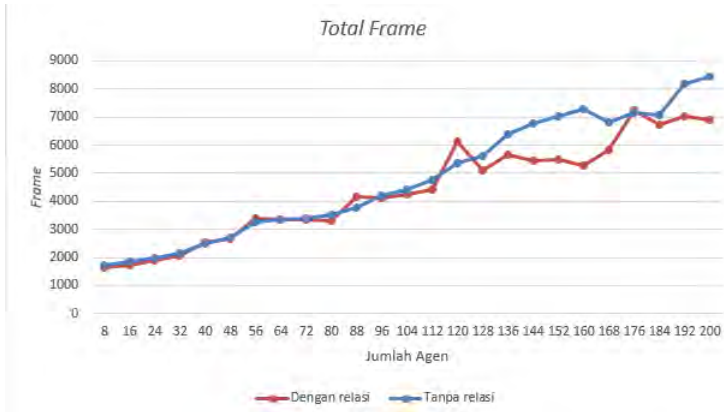
ge mengikuti *leader*-nya. Hal ini dikarenakan ketergantungan agen *follower* terhadap agen *leader* sehingga lebih sedikit ditemui adanya agen *follower* memutar *narrow passage*. Peningkatan jumlah *frame* beserta waktu simulasi yang ditunjukkan pada gambar 4.14 dan 4.16 memperlihatkan perbedaan yang tidak terlalu signifikan hing-

ga jumlah agen mencapai 128 agen. Setelahnya, terjadi perbedaan jumlah baik *frame* dan waktu simulasi dimana simulasi dengan relasi *leader & follower* memiliki nilai lebih rendah. Rata-rata *frame rate* untuk keduanya relatif stabil antara 59 hingga 60.5 fps sebagaimana ditunjukkan pada grafik di gambar 4.17.

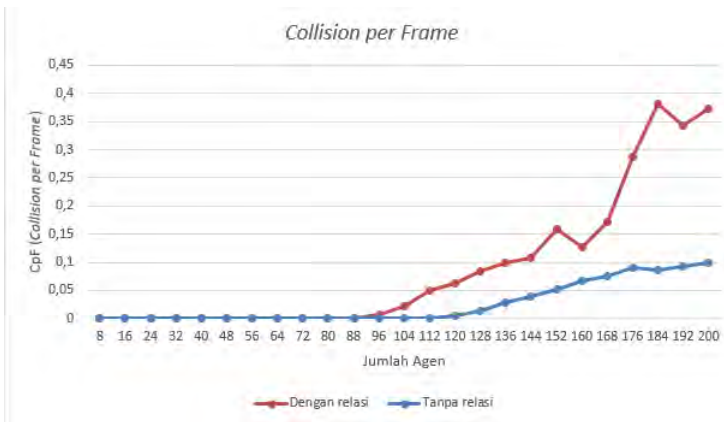
Tidak seperti skenario sebelumnya (*crossroad scenario*), Tidak terjadi penurunan *frame rate* yang signifikan. Hal ini berhubungan dengan data CpF yang dapat dilihat pada gambar 4.15 dimana nilai CpF yang didapat tidak melebihi satu CpF. Jadi pada simulasi ini, jumlah tabrakan yang terjadi tidak terlalu memengaruhi performa. Waktu yang diperlukan pada simulasi dengan relasi *leader-follower* mengalami penurunan dengan rata-rata 7% dibandingkan dengan waktu untuk simulasi agen yang tanpa relasi. Hal ini menunjukkan bahwa untuk skenario ini, relasi *leader-follower* pada RVO membutuhkan waktu yang dibutuhkan untuk kelompok kerumunan agen menuju tujuannya lebih cepat sebagaimana diperlihatkan pada grafik di gambar 4.19.



Gambar 4.13: Grafik jumlah tabrakan yang terjadi pada simulasi *narrow passage scenario*



Gambar 4.14: Grafik jumlah frame pada simulasi *narrow passage scenario*



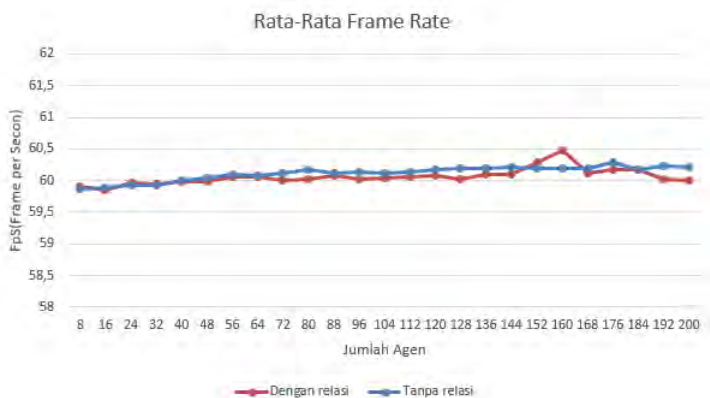
Gambar 4.15: Grafik *collision per frame* untuk simulasi *narrow passage scenario*

4.2.3 Tour Guide Scenario

Skenario pengujian ini memiliki area pengujian yang terluas dibandingkan dua skenario sebelumnya (*crossroad* dan *narrow passage*). Dari data jumlah tabrakan yang diperlihatkan pada gambar

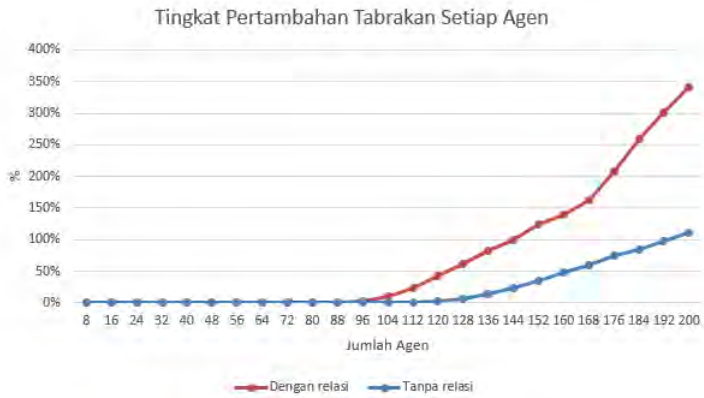


Gambar 4.16: Grafik waktu simulasi *narrow passage scenario*



Gambar 4.17: Grafik rata-rata *frame rate* untuk simulasi *narrow passage scenario*

4.20, agen tanpa relasi mengalami tabrakan paling rendah dari ketiga skenario yang diujikan. Hal ini dikarenakan kebebasan agen dalam bernavigasi beserta kepadatan kerumunan yang relatif rendah sekalipun pada skenario ini banyak terdapat rintangan statis di jalur yang dilalui agen. Berbeda dengan agen yang memiliki relasi



Gambar 4.18: Grafik peningkatan tabrakan agen untuk simulasi *narrow passage scenario*

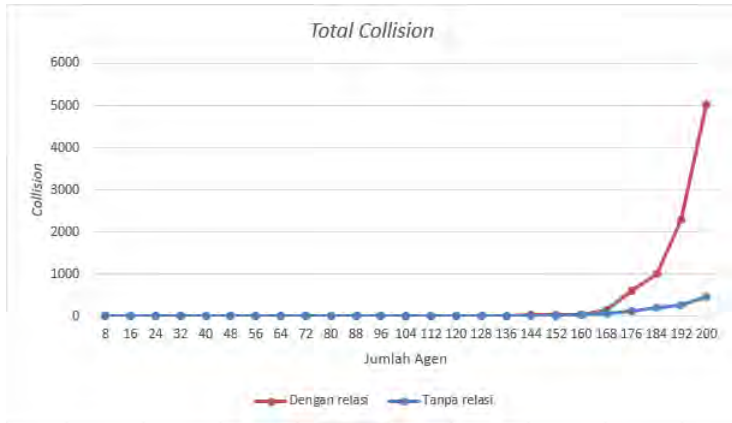


Gambar 4.19: Grafik penurunan waktu simulasi *narrow passage scenario*

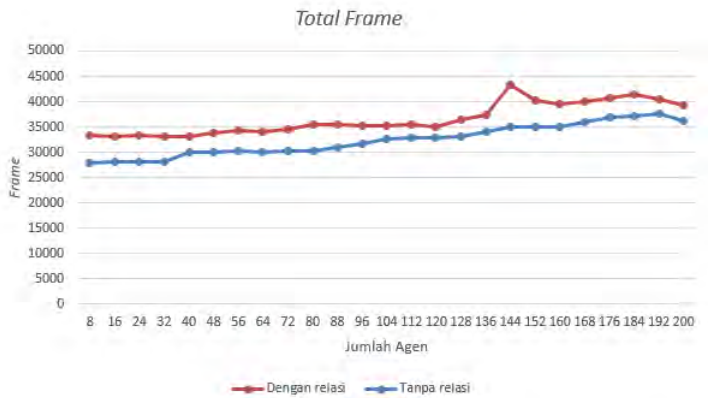
leader & follower yang mana terjadi perbedaan jumlah tabrakan yang signifikan mulai dari jumlah agen sebanyak 176. Tabrakan yang ada, seringkali terjadi pada pertemuan ke dua kelompok agen di ruang yang ditunjukkan pada gambar 3.15e.

Waktu yang diperlukan untuk menyelesaikan tur mengalami peningkatan sebanding dengan penambahan jumlah agen. Sebagaimana ditunjukkan di grafik pada gambar 4.23, Waktu yang dibutuhkan untuk agen dengan relasi *leader & follower* lebih lama dibandingkan dengan agen tanpa relasi. Tidak seperti dua skenario sebelumnya dimana waktu simulasi oleh agen dengan relasi *leader & follower* memiliki waktu yang lebih sedikit. Ini menunjukkan pengaruh rintangan statis yang ada pada skenario ini mengakibatkan gangguan pada gerakan agen *follower* dalam mengikuti leadernya sehingga waktu simulasi menjadi lebih lama. Gangguan ini terjadi karena rintangan di tengah jalannya kelompok mengurangi *valid velocity* pada RVO sehingga gerak agen *follower* dalam mengikuti leadernya sangat terbatas. Berbeda dengan agen tanpa relasi yang lebih bebas dalam memilih pergerakan disetiap siklusnya tanpa perlu mengikuti *leader*. Hal ini terjadi terutama pada saat agen *follower* perlu melakukan *steering* keluar dari area rectangular didepan agen *leader*, rintangan statis seringkali membuat *velocity* referensi menjadi tidak valid sehingga *path* agen *leader* terganggu oleh *follower*. Karena hal inilah grafik pada gambar 4.26 menunjukkan nilai presentase kenaikan waktu simulasi dengan relasi *leader-follower* terhadap simulasi dengan agen tanpa relasi bernilai positif dengan rerata sebesar 13%.

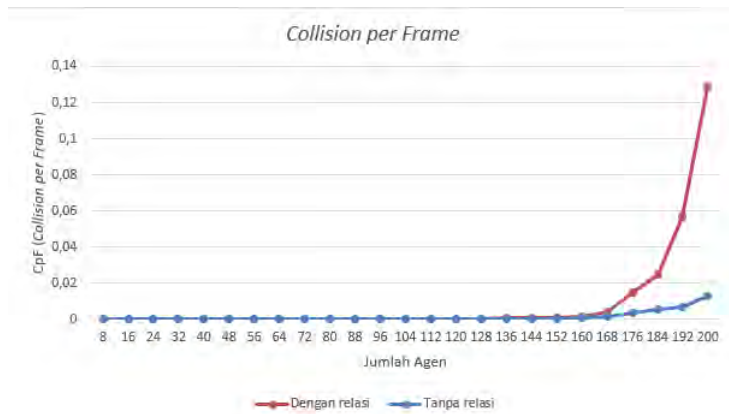
Waktu simulasi yang lama juga mengakibatkan penambahan jumlah *frame* untuk setiap simulasinya sebagaimana dapat dilihat pada gambar 4.21. Rata-rata *frame rate* yang ada pada gambar 4.24 menunjukkan kecendrungan penurunan dengan semakin bertambahnya jumlah agen. Nilai rata-rata *frame rate* agen dengan relasi *leader & follower* lebih rendah dari agen tanpa relasi. Nilai CpF dari simulasi ini bertambah secara eksponensial sebagaimana penambahan jumlah tabrakan.



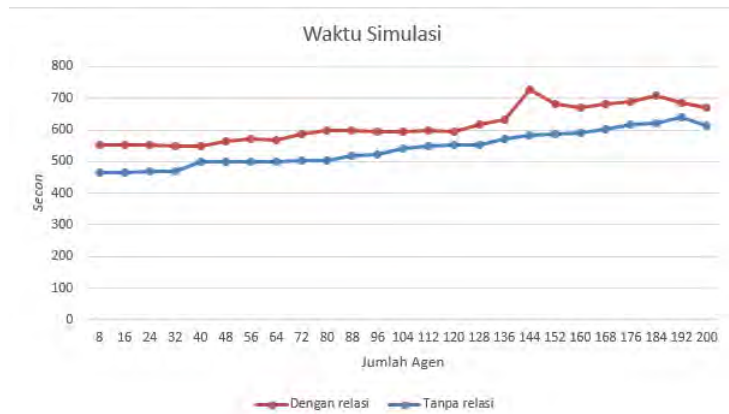
Gambar 4.20: Grafik jumlah tabrakan yang terjadi pada simulasi *tour guide scenario*



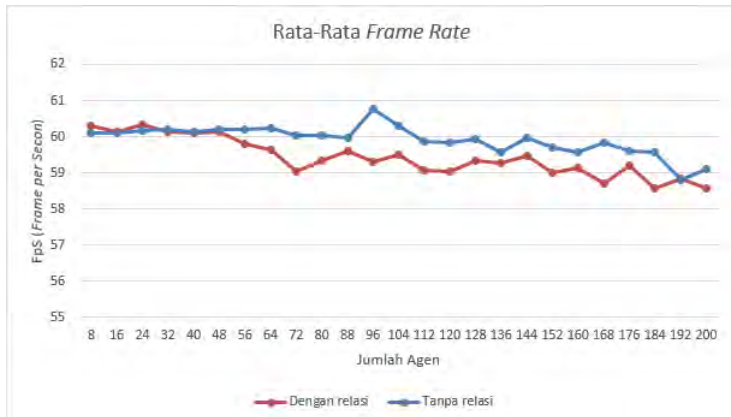
Gambar 4.21: Grafik jumlah frame pada simulasi *tour guide scenario*



Gambar 4.22: Grafik *collision per frame* untuk simulasi *tour guide scenario*



Gambar 4.23: Grafik waktu simulasi *tour guide scenario*



Gambar 4.24: Grafik rata-rata *frame rate* untuk simulasi *tour guide scenario*



Gambar 4.25: Grafik peningkatan tabrakan agen untuk simulasi *tour guide scenario*



Gambar 4.26: Grafik peningkatan waktu simulasi *tour guide scenario*

BAB 5

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian RVO pada simulasi kerumunan dengan relasi *leader & follower* yang sudah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut :

1. Hasil simulasi *crossroad scenario* menggunakan RVO dengan adanya relasi *leader & follower* menghasilkan tabrakan mulai dari jumlah agen sebanyak 200 agen dan meningkat secara eksponensial dengan presentase pertambahan tabrakan hingga 5649% pada jumlah agen sebesar 100 agen. Waktu yang dibutuhkan kelompok agen dengan relasi *leader-follower* lebih cepat dibandingkan dengan kelompok agen tanpa relasi dengan rata-rata penurunan sebesar 18%
2. Hasil simulasi *narrow passage scenario* menggunakan RVO dengan adanya relasi *leader & follower* menghasilkan tabrakan mulai dari jumlah agen sebanyak 88 agen dan meningkat secara eksponensial dengan presentase pertambahan tabrakan hingga 341% pada jumlah agen sebesar 200 agen. Waktu yang dibutuhkan kelompok agen dengan relasi *leader-follower* lebih cepat dibandingkan dengan kelompok agen tanpa relasi dengan rata-rata penurunan sebesar 7%
3. Hasil simulasi *tour guide scenario* menggunakan RVO dengan adanya relasi *leader & follower* menghasilkan tabrakan mulai dari jumlah agen sebanyak 120 agen dan meningkat secara eksponensial dengan presentase pertambahan tabrakan hingga 192% pada jumlah agen sebesar 200 agen. Waktu yang dibutuhkan kelompok agen dengan relasi *leader-follower* lebih lama dibandingkan dengan kelompok agen tanpa relasi dengan rata-rata peningkatan sebesar 13%
4. Rintangan statis lebih memengaruhi waktu simulasi pada navigasi multiagen menggunakan RVO dengan relasi *leader & follower*. Keberadaan rintangan statis seperti pada *tour guide scenario* membuat waktu simulasi pada agen dengan relasi

leader & follower lebih lama dari pada agen tanpa relasi.

5.2 Saran

Demi pengembangan lebih lanjut mengenai tugas akhir ini, disarankan beberapa langkah lanjutan sebagai berikut :

1. Pengujian simulasi dengan jumlah agen simulasi lebih banyak pada skenario dan kondisi yang berbeda menggunakan komputer dengan spesifikasi yang lebih tinggi.
2. Implementasi parameter prioritas penghindaran RVO terutama bagi agen *leader* untuk menggantikan sistem *steering* agen *follower* yang berada didepan *leader*.

DAFTAR PUSTAKA

- [1] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," University of North California, 2008. (Dikutip pada halaman 1, 2, 5, 7, 8, 9, 13, 15, 16, 22).
- [2] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," University of North California, 2011. (Dikutip pada halaman 1).
- [3] J. van den Berg, S. Pati, J. Sewall, D. Manocha, and M. Lin, "Interactive navigation of multiple agents in crowded environment," University of North California, 2008. (Dikutip pada halaman 1, 13, 15, 16).
- [4] S. J. Guy, S. Kim, M. C. Lin, and D. Manocha, "Simulating heterogeneous crowd behaviors using personality trait theory," University of North California, 2011. (Dikutip pada halaman 1).
- [5] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacles," IEEE Transaction on Robotic, vol. 27, 2011. (Dikutip pada halaman 1).
- [6] J. Snape, S. J. Guy, D. Vembar, A. L. nad Ming C. Lin, and D. Manocha, "Reciprocal collision avoidance and navigation for video games," Intel Software Network, 2008. (Dikutip pada halaman 1).
- [7] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," vol. 17, no. 7, p. 760772, 1998. (Dikutip pada halaman 5).
- [8] R. Legget, "Real-time crowd simulation: a review," 2004. (Dikutip pada halaman 9).
- [9] A. Kurniawati, S. M. Nugroho, and M. Hariadi, "Simulasi pergerakan pengunjung mall menggunakan potential field," vol. 5, January 2010. (Dikutip pada halaman 10).

- [10] C. W. Reynolds, “Steering behaviors for autonomous characters,” 1999. (Dikutip pada halaman 10, 11).
- [11] K. Aspelin, “Establishing pedestrian walking speeds,” January 2005. (Dikutip pada halaman 34).

LAMPIRAN

A. Hasil simulasi *crossroad scenario*

Tabel A1: Data simulasi *crossroad scenario leader-follower*

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	Collision per Frame	Waktu simulasi	Frame rate rata-rata
1	10	0	2987	0	49,73201	60,06191988
2	20	0	2979	0	49,45003	60,24263282
3	30	0	2953	0	49,17238	60,05403847
4	40	0	3004	0	50,04134	60,03036689
5	50	0	3086	0	50,7066	60,8599275
6	60	0	3096	0	51,55376	60,05381567
7	70	0	3203	0	53,33751	60,05154721
8	80	0	3196	0	53,21359	60,05984561
9	90	0	3188	0	53,06426	60,07810153
10	100	3	3310	0,000906344	55,10785	60,06403806
11	110	15	3325	0,004511278	55,35614	60,06560429
12	120	19	3310	0,005740181	55,11403	60,05730301
13	130	351	3218	0,109073959	53,79115	59,82396733
14	140	526	3273	0,16070883	53,87558	60,75108611
15	150	2095	3194	0,655917345	53,22179	60,01301347
16	160	6813	3298	2,065797453	54,87904	60,09580343
17	170	10435	3415	3,055636896	56,8778	60,04100018
18	180	25960	3721	6,976619188	63,30436	58,77952166
19	190	55871	3854	14,49688635	67,71167	56,91781047
20	200	113356	4045	28,023733	74,43262	54,34445274

Tabel A2: Data simulasi *crossroad scenario* tanpa relasi

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	Collision per Frame	Waktu simulasi	Frame rate rata-rata
1	10	0	2815	0	46,55662	60,46401135
2	20	0	2865	0	47,6581	60,11569912
3	30	0	2888	0	48,04103	60,11528063
4	40	0	2914	0	48,4766	60,11147646
5	50	0	3488	0	57,99089	60,14737832
6	60	0	3635	0	60,4581	60,12428442
7	70	0	3695	0	61,41816	60,16135944
8	80	0	3972	0	66,00547	60,1768308
9	90	0	3980	0	66,19099	60,12902965
10	100	0	4287	0	71,23515	60,18096403

Dilanjutkan di halaman berikutnya

Tabel A2 – Lanjutan dari halaman sebelumnya

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	Collision per Frame	Waktu simulasi	Frame rate rata-rata
11	110	0	4016	0	66,79388	60,12526896
12	120	22	4689	0,004691832	77,88213	60,2063657
13	130	105	4708	0,022302464	78,41222	60,04166187
14	140	571	4600	0,124130435	76,46093	60,16144454
15	150	2111	5357	0,394063842	88,95283	60,22292939
16	160	3789	5492	0,6899126	91,25867	60,18058339
17	170	5749	5013	1,146818272	83,33652	60,15369972
18	180	6822	5146	1,325689856	85,51659	60,17545835
19	190	6270	4742	1,322226908	78,87943	60,11706728
20	200	6631	5140	1,290077821	85,44038	60,15890847

Tabel A3: Data perbandingan RVO dengan relasi *leader-follower* dan tanpa relasi pada *crossroad scenario*

No.	Jumlah Agen	Tingkat Pertambahan Tabrakan Setiap Agen (Leader-Follower)	Tingkat Pertambahan Tabrakan Setiap Agen Tanpa Relasi	Penurunan waktu simulasi
1	10	0%	0%	-7%
2	20	0%	0%	-4%
3	30	0%	0%	-2%
4	40	0%	0%	-3%
5	50	0%	0%	13%
6	60	0%	0%	15%
7	70	0%	0%	13%
8	80	0%	0%	19%
9	90	0%	0%	20%
10	100	0%	0%	23%
11	110	2%	0%	17%
12	120	3%	2%	29%
13	130	23%	8%	31%
14	140	48%	36%	30%
15	150	138%	128%	40%
16	160	396%	268%	40%
17	170	734%	451%	32%
18	180	1494%	636%	26%
19	190	2963%	777%	14%
20	200	5649%	903%	13%

B. Hasil simulasi *narrow passage scenario*

Tabel B1: Data simulasi *narrow passage leader-follower*

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	Collision per Frame	Waktu simulasi	Frame rate rata-rata
1	8	0	1653	0	27,58945	59,91420634
2	16	0	1708	0	28,54145	59,84279005
3	24	0	1910	0	31,85644	59,95647976
4	32	0	2081	0	34,71583	59,94383542
5	40	0	2524	0	42,07607	59,98659095
6	48	0	2685	0	44,75563	59,99245235
7	56	0	3411	0	56,79572	60,05734235
8	64	0	3332	0	55,4856	60,051617
9	72	0	3333	0	55,54585	60,00448278
10	80	0	3313	0	55,18908	60,02999144
11	88	7	4164	0,001681076	69,3174	60,0714972
12	96	28	4105	0,00682095	68,39043	60,02301784
13	104	97	4238	0,022888155	70,58228	60,04339899
14	112	221	4403	0,05019305	73,31213	60,05827412
15	120	384	6116	0,062786135	101,8154	60,06949833
16	128	427	5125	0,083317073	85,38844	60,01983407
17	136	562	5670	0,099118166	94,35854	60,08995052
18	144	586	5465	0,107227813	90,9214	60,10686153
19	152	867	5486	0,158038644	91,00878	60,27989827
20	160	673	5293	0,127149065	87,52996	60,4707234
21	168	1002	5813	0,172372269	96,69022	60,11983425
22	176	2080	7231	0,287650394	120,1688	60,17368901
23	184	2576	6737	0,382366038	111,953	60,17703858
24	192	2413	7024	0,343536446	117,0089	60,02962168
25	200	2582	6920	0,373121387	115,3216	60,00610467

Tabel B2: Data simulasi *narrow passage* tanpa relasi

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	Collision per Frame	Waktu simulasi	Frame rate rata-rata
1	8	0	1736	0	29,00096	59,86008739
2	16	0	1852	0	30,92348	59,88976661
3	24	0	1965	0	32,78578	59,93452039
4	32	0	2132	0	35,57479	59,9300797
5	40	0	2478	0	41,3042	59,99389893
6	48	0	2710	0	45,13215	60,04588747
7	56	0	3254	0	54,14796	60,09460006

Dilanjutkan di halaman berikutnya

Tabel B2 – Lanjutan dari halaman sebelumnya

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	<i>Collision per Frame</i>	Waktu simulasi	<i>Frame rate rata-rata</i>
8	64	0	3357	0	55,87197	60,08379515
9	72	0	3412	0	56,76433	60,10817004
10	80	0	3540	0	58,82732	60,17612225
11	88	2	3778	0,000529381	62,84985	60,11151976
12	96	0	4206	0	69,93931	60,1378538
13	104	5	4412	0,001133273	73,39017	60,11704292
14	112	6	4745	0,001264489	78,89557	60,14279382
15	120	24	5341	0,004493541	88,76124	60,17266095
16	128	71	5613	0,012649207	93,24992	60,19308113
17	136	182	6377	0,028540066	105,9543	60,18632561
18	144	273	6793	0,040188429	112,814	60,21415782
19	152	364	7018	0,051866629	116,5877	60,19502915
20	160	487	7294	0,066767206	121,1696	60,19661697
21	168	517	6816	0,075850939	113,2456	60,18776888
22	176	644	7178	0,089718585	119,0626	60,28761341
23	184	604	7071	0,085419318	117,4983	60,17959409
24	192	757	8186	0,092474957	135,9229	60,22531891
25	200	837	8458	0,098959565	140,4892	60,20391603

Tabel B3: Data perbandingan RVO dengan relasi *leader-follower* dan tanpa relasi pada *narrow passage scenario*

No.	Jumlah Agen	Tingkat Pertambahan Tabrakan Setiap Agen (<i>Leader-Follower</i>)	Tingkat Pertambahan Tabrakan Setiap Agen Tanpa Relasi	Penurunan waktu simulasi
1	8	0%	0%	5%
2	16	0%	0%	8%
3	24	0%	0%	3%
4	32	0%	0%	2%
5	40	0%	0%	-2%
6	48	0%	0%	1%
7	56	0%	0%	-5%
8	64	0%	0%	1%
9	72	0%	0%	2%
10	80	0%	0%	6%
11	88	1%	0%	-10%
12	96	3%	0%	2%
13	104	10%	1%	4%
14	112	23%	1%	7%

Dilanjutkan di halaman berikutnya

Tabel B3 – Lanjutan dari halaman sebelumnya

No.	Jumlah Agen	Tingkat Pertambahan Tabrakan Setiap Agen (<i>Leader-Follower</i>)	Tingkat Pertambahan Tabrakan Setiap Agen Tanpa Relasi	Penurunan waktu simulasi
15	120	43%	2%	-15%
16	128	61%	5%	8%
17	136	82%	13%	11%
18	144	100%	23%	19%
19	152	125%	34%	22%
20	160	140%	48%	28%
21	168	161%	60%	15%
22	176	208%	74%	-1%
23	184	260%	85%	5%
24	192	301%	98%	14%
25	200	341%	111%	18%

C. Hasil simulasi *tour guide scenario*Tabel C1: Data simulasi *tour guide leader-follower*

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	Collision per Frame	Waktu simulasi	Frame rate rata-rata
1	8	0	33236	0	551,1464	60,3033967
2	16	0	33144	0	551,30017	60,11969849
3	24	0	33285	0	551,6564	60,33647031
4	32	0	33098	0	550,3661	60,13815168
5	40	0	33038	0	549,6413	60,10829244
6	48	0	33847	0	562,9611	60,12315949
7	56	0	34188	0	571,7884	59,79134939
8	64	0	33956	0	569,2461	59,65082589
9	72	0	34613	0	586,3244	59,03387272
10	80	0	35488	0	598,2356	59,32111028
11	88	0	35555	0	596,6003	59,59601428
12	96	0	35156	0	592,8502	59,29997156
13	104	0	35288	0	593,0233	59,50525047
14	112	0	35407	0	599,2561	59,08492212
15	120	1	35088	2,84998E-05	594,2021	59,05061594
16	128	15	36527	0,000410655	615,5782	59,33770884
17	136	21	37428	0,000561077	631,5058	59,26786421
18	144	25	43364	0,000576515	729,1109	59,47517723
19	152	34	40168	0,000846445	680,7756	59,00328978

Dilanjutkan di halaman berikutnya

Tabel C1 – Lanjutan dari halaman sebelumnya

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	<i>Collision per Frame</i>	Waktu simulasi	<i>Frame rate rata-rata</i>
20	160	51	39603	0,001287781	669,563	59,14753354
21	168	159	39952	0,003979776	680,5589	58,7046911
22	176	606	40756	0,014868976	688,3231	59,2105655
23	184	1019	41511	0,02454771	708,7504	58,56927911
24	192	2281	40376	0,056493957	686,3779	58,8247378
25	200	5043	39246	0,128497172	670,1158	58,5659971

Tabel C2: Data simulasi *tour guide* tanpa relasi

No.	Jumlah Agen	Jumlah collision	Jumlah Frame	<i>Collision per Frame</i>	Waktu simulasi	<i>Frame rate rata-rata</i>
1	8	0	27998	0	465,7468	60,11420798
2	16	0	28023	0	466,1366	60,11757069
3	24	0	28103	0	467,2003	60,15193055
4	32	0	28225	0	468,7239	60,2166862
5	40	0	30024	0	499,3855	60,1218898
6	48	0	29991	0	498,256	60,19194952
7	56	0	30144	0	500,7848	60,19352025
8	64	0	30078	0	499,2623	60,2448853
9	72	0	30201	0	503,0801	60,0321897
10	80	0	30195	0	502,9827	60,03188579
11	88	0	30998	0	516,8151	59,97889768
12	96	0	31777	0	523,0111	60,75779271
13	104	0	32740	0	542,9154	60,30405474
14	112	3	32811	9,14328E-05	548,1544	59,85722271
15	120	2	32982	6,06391E-05	551,214	59,83520012
16	128	5	33088	0,000151112	552,0307	59,93869544
17	136	7	33957	0,000206143	569,8861	59,58559088
18	144	14	34944	0,000400641	582,6848	59,97067368
19	152	12	34977	0,000343083	585,9144	59,69643347
20	160	28	35131	0,000797017	589,7003	59,57432954
21	168	59	36021	0,001637933	602,1178	59,82384178
22	176	131	36815	0,003558332	617,5283	59,61670097
23	184	204	37085	0,005500876	622,4871	59,57553177
24	192	259	37707	0,006868751	641,2587	58,80154141
25	200	453	36162	0,012526962	611,9167	59,09627895

Tabel C3: Data perbandingan RVO dengan relasi *leader-follower* dan tanpa relasi pada *crossroad scenario*

No.	Jumlah Agen	Tingkat Pertambahan Tabrakan Setiap Agen (Leader-Follower)	Tingkat Pertambahan Tabrakan Setiap Agen tanpa relasi	Kenaikan waktu simulasi
1	8	0%	0%	18%
2	16	0%	0%	18%
3	24	0%	0%	18%
4	32	0%	0%	17%
5	40	0%	0%	10%
6	48	0%	0%	13%
7	56	0%	0%	14%
8	64	0%	0%	14%
9	72	0%	0%	17%
10	80	0%	0%	19%
11	88	0%	0%	15%
12	96	0%	0%	13%
13	104	0%	0%	9%
14	112	0%	0%	9%
15	120	0%	0%	8%
16	128	1%	1%	12%
17	136	2%	1%	11%
18	144	3%	1%	25%
19	152	4%	2%	16%
20	160	5%	2%	14%
21	168	9%	4%	13%
22	176	24%	7%	11%
23	184	47%	12%	14%
24	192	95%	17%	7%
25	200	192%	25%	10%

D. Jumlah agen yang tersesat pada penggunaan relasi *leader - follower*

Tabel D: Data jumlah agen yang tersesat pada penggunaan relasi *leader - follower*

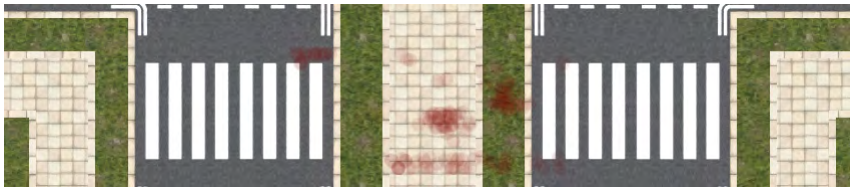
No.	Jumlah Agen	Crossroad scenario	Narrow passage scenario	Tour guide scenario
1	8	0	0	0
2	16	0	0	0
3	24	0	0	0

Dilanjutkan di halaman berikutnya

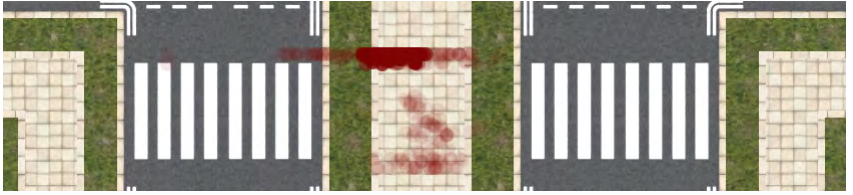
Tabel D – Lanjutan dari halaman sebelumnya

No.	Jumlah Agen	Crossroad scenario	Narrow passage scenario	Tour guide scenario
4	32	0	0	0
5	40	0	0	0
6	48	0	0	0
7	56	0	0	0
8	64	0	0	0
9	72	0	0	0
10	80	0	0	0
11	88	0	0	0
12	96	0	0	0
13	104	0	0	0
14	112	0	0	0
15	120	0	0	0
16	128	0	0	0
17	136	0	0	1
18	144	0	0	1
19	152	0	0	1
20	160	0	0	2
21	168	2	0	2
22	176	2	0	3
23	184	6	0	2
24	192	11	0	4
25	200	16	0	4

E. *Collision heatmap*



Gambar E1: *Heatmap* skenario *crossroad* 120 agen dengan relasi



Gambar E2: *Heatmap* skenario *crossroad* 130 agen dengan relasi



Gambar E3: *Heatmap* skenario *crossroad* 140 agen dengan relasi



Gambar E4: *Heatmap* skenario *crossroad* 150 agen dengan relasi



Gambar E5: *Heatmap* skenario *crossroad* 160 agen dengan relasi



Gambar E6: *Heatmap* skenario *crossroad* 170 agen dengan relasi



Gambar E7: *Heatmap* skenario *crossroad* 180 agen dengan relasi



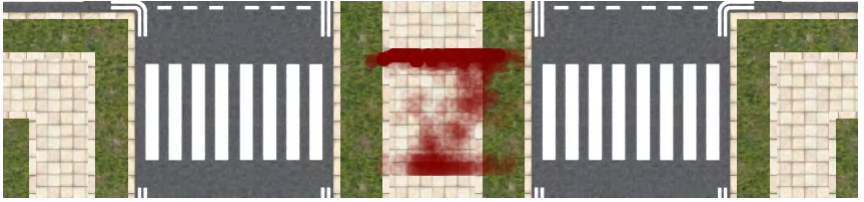
Gambar E8: *Heatmap* skenario *crossroad* 190 agen dengan relasi



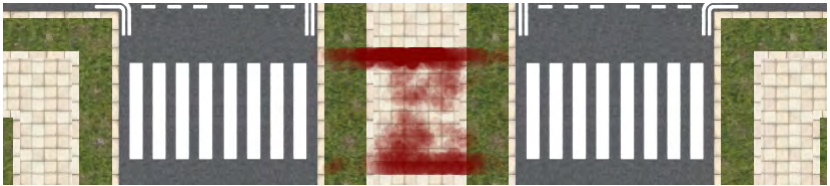
Gambar E9: *Heatmap* skenario *crossroad* 200 agen dengan relasi



Gambar E10: *Heatmap* skenario *crossroad* 140 agen tanpa relasi



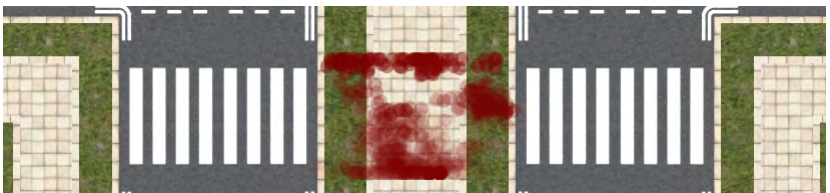
Gambar E11: *Heatmap* skenario *crossroad* 150 agen tanpa relasi



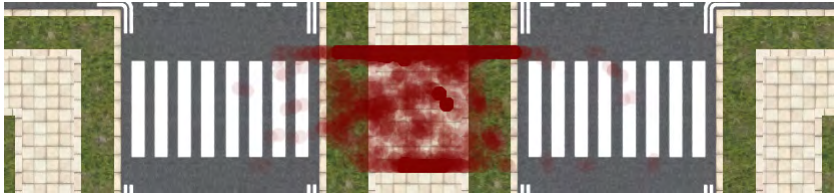
Gambar E12: *Heatmap* skenario *crossroad* 160 agen tanpa relasi



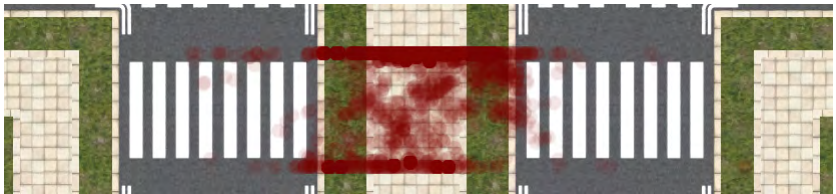
Gambar E13: *Heatmap* skenario *crossroad* 170 agen tanpa relasi



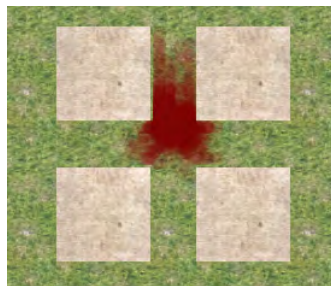
Gambar E14: *Heatmap* skenario *crossroad* 180 agen tanpa relasi



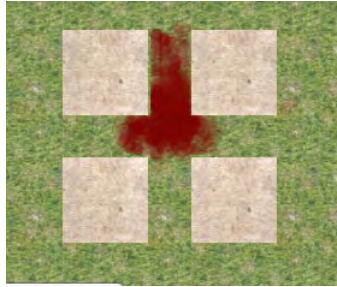
Gambar E15: *Heatmap* skenario *crossroad* 190 agen tanpa relasi



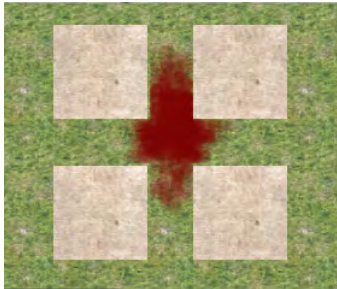
Gambar E16: *Heatmap* skenario *crossroad* 200 agen tanpa relasi



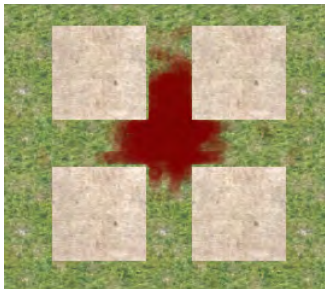
Gambar E17: *Heatmap* skenario *narrow passage* 128 agen dengan relasi



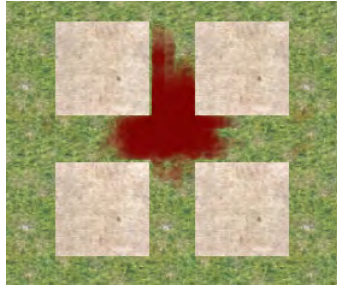
Gambar E18: *Heatmap* skenario *narrow passage* 136 agen dengan relasi



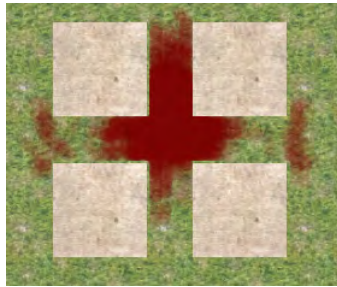
Gambar E19: *Heatmap* skenario *narrow passage* 144 agen dengan relasi



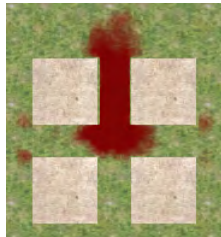
Gambar E20: *Heatmap* skenario *narrow passage* 152 agen dengan relasi



Gambar E21: *Heatmap* skenario *narrow passage* 160 agen dengan relasi



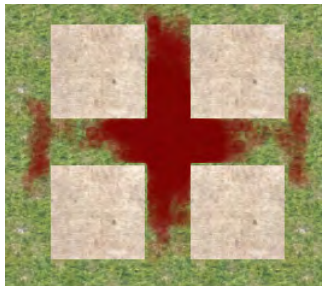
Gambar E22: *Heatmap* skenario *narrow passage* 168 agen dengan relasi



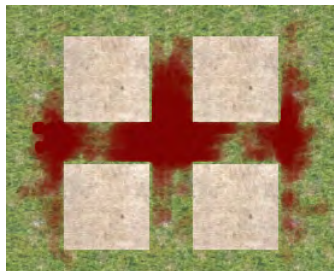
Gambar E23: *Heatmap* skenario *narrow passage* 176 agen dengan relasi



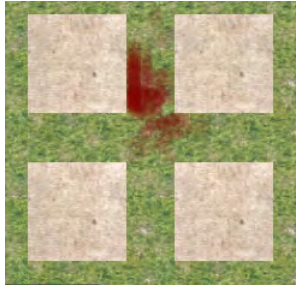
Gambar E24: *Heatmap* skenario *narrow passage* 184 agen dengan relasi



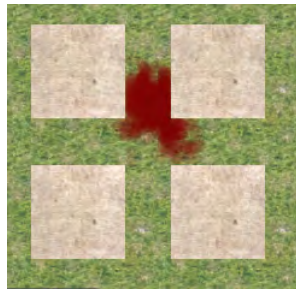
Gambar E25: *Heatmap* skenario *narrow passage* 192 agen dengan relasi



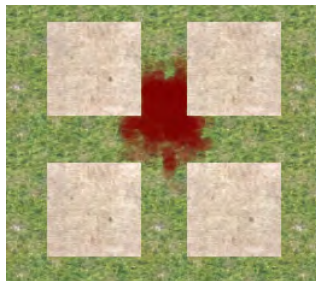
Gambar E26: *Heatmap* skenario *narrow passage* 200 agen dengan relasi



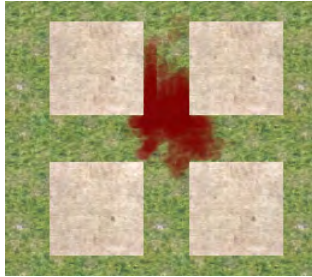
Gambar E27: *Heatmap* skenario *narrow passage* 136 agen tanpa relasi



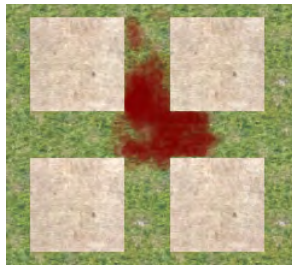
Gambar E28: *Heatmap* skenario *narrow passage* 144 agen tanpa relasi



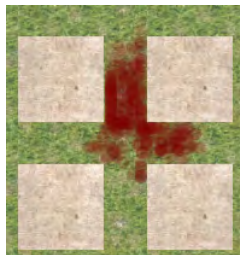
Gambar E29: *Heatmap* skenario *narrow passage* 152 agen tanpa relasi



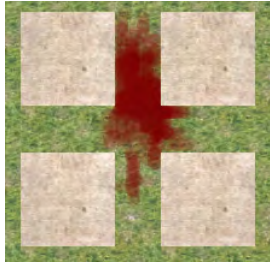
Gambar E30: *Heatmap* skenario *narrow passage* 160 agen tanpa relasi



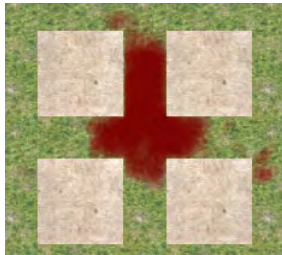
Gambar E31: *Heatmap* skenario *narrow passage* 168 agen tanpa relasi



Gambar E32: *Heatmap* skenario *narrow passage* 176 agen tanpa relasi



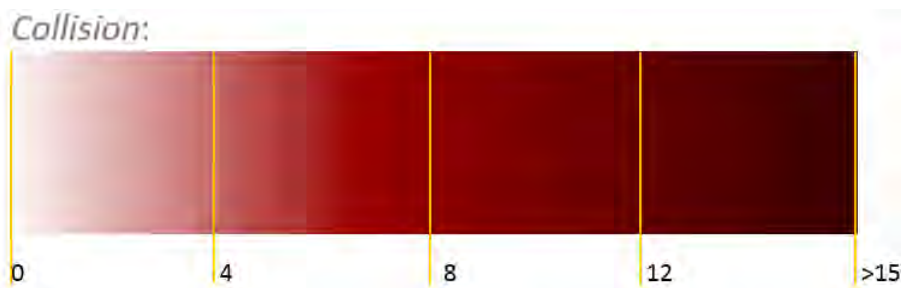
Gambar E33: *Heatmap* skenario *narrow passage* 184 agen tanpa relasi



Gambar E34: *Heatmap* skenario *narrow passage* 192 agen tanpa relasi



Gambar E35: *Heatmap* skenario *narrow passage* 200 agen tanpa relasi



Gambar E36: *Collision heatmap legend*

Halaman ini sengaja dikosongkan

BIOGRAFI PENULIS



Moch Fachri, lahir pada 18 Januari 1992 di Surabaya, Jawa Timur. Penulis lulus dari SMP Negeri 1 Surabaya pada tahun 2007 kemudian melanjutkan pendidikan ke SMA Negeri 2 Surabaya hingga akhirnya lulus pada tahun 2010. Penulis kemudian melanjutkan pendidikan strata satu di Jurusan Teknik Elektro ITS Surabaya bidang studi Teknik Komputer dan Telematika. Saat di kuliah penulis aktif menjadi kepala departemen syiar kajian Islam Himatektro ITS 2012/2013. Penulis juga aktif menjadi Asisten laboratorium B201 (Telematika) hingga saat ini dan pernah menjabat sebagai koordinator *game*

development Lab B201 dan asisten praktikum rangkaian digital periode 2013/2014. Selama masa kuliah penulis aktif dalam mengikuti ajang perlombaan seperti PKM (Program Kreativitas Mahasiswa) serta kegiatan kerohanian kampus. Penulis sangat tertarik dengan segala hal yang berhubungan dengan komputer, dan berencana mendalami cabang ilmu komputer lain selain teknologi game.

Halaman ini sengaja dikosongkan